

# PEMBUATAN ANALISA KINERJA KEPEGAWAIAN PENS-ITS MENGGUNAKAN JSP

Alfika Zakiyatur Rohmah<sup>1</sup>, Edi Satriyanto<sup>2</sup>, S. Si, M.Si, Wiratmoko Yuwono<sup>2</sup>, ST

<sup>1</sup> Mahasiswa, <sup>2</sup> Dosen Pembimbing

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember Kampus ITS Keputih Sukolilo Surabaya 60111, Indonesia

Telp: +62-31-5947280 Fax: +62-31-5946114

Email: [al\\_vix\\_z@yahoo.co.id](mailto:al_vix_z@yahoo.co.id)

## ABSTRAK

Kinerja pegawai merupakan hasil kerja sebagai akibat suatu tindakan seseorang dalam melaksanakan tugas dengan dilandasi oleh kemampuan, sikap dan motivasinya. Kinerja dalam menjalankan fungsinya tidak berdiri sendiri melainkan berhubungan dengan faktor individu, organisasi dan lingkungan external.

Kemajuan teknologi telah menjanjikan suatu hasil pekerjaan yang efektif dan efisien serta mampu memenangkan kompetitif, dengan sistem teknologi yang semakin pesat, sistem-sistem informasi-pun begitu mudah di peroleh, semakin akurat dan cepat, tetapi bagaimanapun pesatnya kemajuan teknologi, peranan manusia masih menduduki peringkat terpenting, karena teknologi hanyalah berupa alat bantu berupa hardware, sedangkan manusia itu sendiri yang mengoptimalkan.

Proyek ini mencoba membangun suatu aplikasi dengan menggunakan pemrograman JSP dan database MySQL yang diimplementasikan dalam bentuk web yang informatif, yang berguna untuk membantu memberikan analisa kinerja kepegawaian kepada pegawai setelah memasukkan kriteria-kriteria yang mereka inginkan.

*Kata kunci : Kinerja Kepegawaian, JSP dan MySQL*

## ABSTRACT

*Performance of Officer represent result of job as effect of an action of somebody in executing duty based only by ability, attitude and motivation. Performance in running his function is not selfsupporting but relate to individual factor, external environment and organization.*

*Technological progress have made a promise an effective work result and efisien and also able to win kompetitif, with system of technology which is fast progressively, information system so easy to in obtaining, accurate progressively and quickly, but however its his fast is technological progress, role of human being still occupy all important rank, because technology only in the form of assistive appliance in the form of hardware, while human being optimal itself.*

*This project tried to develop an application with used JSP and MySQL that implemented in the form of web that was informative, what is good for assisting to give officer performance analysis to officer after user put criteria that were wanted by them.*

*Keyword : Officer Performance, JSP, MySQL*

## I. PENDAHULUAN

### 1.1 LATAR BELAKANG

Kinerja merupakan tanda berhasil atau tidaknya seseorang atau organisasi dalam melaksanakan pekerjaan nyata yang ditetapkan dengan standar tertinggi dari orang tersebut, yang melampaui apa yang diminta atau diharapkan. Selain itu kinerja pegawai yang baik terlihat dari hasil menyelesaikan pekerjaan, kinerja juga berkaitan dengan tingkat pengetahuan dan kemampuan dalam mengatasi masalah.

Kinerja pegawai meliputi data pribadi, data absensi, daily activity, kenaikan pangkat pegawai (DP3) dan lain-lain. Sebuah sistem informasi manajemen kepegawaian yang terkomputerisasi dapat digunakan dalam manajemen sumber daya manusia secara terpadu. Selain untuk manajemen pegawai yang bersifat pencatatan, data kepegawaian juga digunakan bagi pihak eksekutif sebuah institusi untuk meng-analisa kinerja sehingga hasil analisa tersebut dapat dilaporkan kepada masing-masing pegawai.

Untuk itu dalam proyek akhir akan dilakukan pembuatan aplikasi analisa kinerja kepegawaian PENS-ITS berbasis JSP yang memberikan informasi mengenai kinerja kepegawaian, khususnya mengenai kinerja absensi, kinerja daily activity dan kenaikan pangkat pegawai (DP3). Laporan proyek akhir ini adalah sebagai sarana untuk melaporkan informasi analisa kinerja kepegawaian PENS-ITS.

### 1.2 TUJUAN

Poyek akhir ini bertujuan untuk membuat laporan kinerja kepegawaian dimana akan memudahkan bagi pegawai untuk mendapatkan informasi mengenai hasil kinerja kepegawaian berupa absensi, daily activity dan kenaikan pangkat pegawai (DP3).

### 1.3 RUANG LINGKUP PERMASALAHAN

Dengan pertimbangan waktu dan kemampuan kami sebagai penyusun dalam penelitian ini membatasi masalah dalam batas-batas tertentu, guna menghasilkan sebuah hasil yang tidak terlalu biasa permasalahannya. Pembahasan dibatasi pada item-item yang tersebut di bawah ini :

1. Sistem dibangun menggunakan teknologi JSP
2. Bahasa pemrograman menggunakan JSP
3. Database yang digunakan adalah MySQL

### 1.4 PERUMUSAN MASALAHAN

Permasalahan yang ditangani adalah sebagai berikut:

- Bagaimana membuat informasi absensi, kenaikan pangkat pegawai (DP3), dan daily activity.

- Bagaimana membuat informasi analisa kinerja absensi, daily activity dan kenaikan pangkat pegawai (DP3).

### 1.5 BATASAN MASALAH

Adapun batasan masalah dalam proyek akhir adalah :

- Hanya membahas pada lingkup Informasi absensi, informasi daily activity dan informasi kenaikan pangkat pegawai (DP3) dan tidak membahas diluar masalah tersebut.
- Tidak membahas kenaikan pangkat pegawai.
- 

### 1.6 METODOLOGI

Metodologi yang digunakan dalam pembuatan Proyek Akhir ini adalah sebagai berikut:

- **Studi Literatur**  
Pada tahap ini dilakukan pemahaman tentang materi-materi yang berhubungan dengan proyek akhir yaitu Java Server Pages (JSP), Konsep Basis Data dan My SQL.
- **Perancangan Sistem.**  
Pada tahap ini dilakukan perancangan terhadap apa saja yang akan dikerjakan serta gambaran dari proyek akhir .Bagaimana tentang desain sistem, alur dari sistem dan desain interface dari program yang kita buat.
- **Pembuatan Perangkat Lunak**  
Hasil dari perancangan sistem ini dengan database Xampp (phpMyAdmin) yang terdistribusi diimplementasikan dalam sebuah perangkat lunak yaitu Pembuatan Analisa Kinerja Kepegawaian PENS-ITS menggunakan JSP.
- **Pengujian Perangkat Lunak dan Perbaikan**  
Pada tahap ini akan dilakukan berbagai uji coba untuk mengukur tingkat keberhasilan aplikasi yang sudah dibangun, juga melakukan perbaikan dan evaluasi untuk meminimalisir kesalahan-kesalahan yang ada.
- **Analisa dan Kesimpulan**  
Dari hasil integrasi dan pengujian sistem secara keseluruhan dilakukan analisa terhadap program yang telah kita buat dan memberikan kesimpulan atas analisa tersebut.
- **Penyusunan Laporan Proyek Akhir**  
Tahap ini digunakan untuk membuat laporan dari semua langkah - langkah yang telah dikerjakan dalam menyelesaikan aplikasi ini. Di dalamnya termasuk dasar teori, metode - metode yang digunakan, perancangan dan pembuatan sistem serta hasil evaluasi dari aplikasi yang telah dibuat.

### 1.7 SISTEMATIKA PEMBAHASAN

Sistematika studi ini dibagi menjadi beberapa bab bahasan yang meliputi :

- **BAB I PENDAHULUAN**  
Membahas tentang latar belakang, tujuan, perumusan masalah, batasan masalah dan sistematika pembahasan.
- **BAB II TEORI PENUNJANG**  
Berisikan pendahuluan, membahas teori JSP, MySQL.

### ▪ **BAB III PERENCANAAN DAN PEMBUATAN**

Berisikan desain sistem tentang pembuatan laporan analisa kinerja kepegawaian berbasis JSP, seperti diagram sistem, daftar tabel beserta fungsinya, relasi antar tabel, desain menu serta desain user interface.

### ▪ **BAB IV PENGUJIAN DAN ANALISA**

Menguji software untuk mengetahui tingkat kesuksesan sistem serta analisa software.

### ▪ **BAB V PENUTUP**

Berisi kesimpulan dan saran.

### **DAFTAR PUSTAKA**

Berisi referensi-referensi yang berhubungan dengan proyek akhir ini, sehingga diharapkan membantu memudahkan pemahaman dalam mempelajari buku laporan.

### **LAMPIRAN**

Berisi keterangan listing program yang digunakan dalam pembuatan program analisa kinerja kepegawaian PENS-ITS berbasis JSP.

### **TENTANG PENULIS**

Berisi tentang biografi penulis, disertai identitas diri sehingga memudahkan pembaca bila ingin berdiskusi tentang isi buku maupun ilmu yang terkait.

## **II. TEORI PENUNJANG**

### **2.1 Java Server Pages (JSP)**

#### **2.1.1 Pengertian JSP**

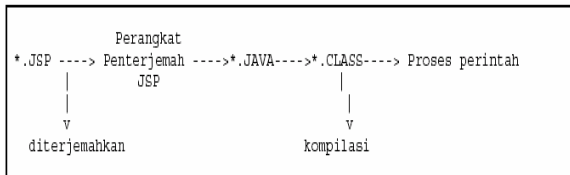
JSP adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, serta merupakan bagian teknologi J2EE (Java 2 Enterprise Edition). JSP sangat sesuai dan tangguh untuk menangani presentasi di web. Sedangkan J2EE merupakan platform Java untuk pengembangan sistem aplikasi enterprise dengan dukungan API (Application Programming Inteface) yang lengkap dan portabilitas serta memberikan sarana untuk membuat suatu aplikasi yang memisahkan antara business logic (sistem), presentasi dan data.

JSP merupakan bagian dari J2EE dan khususnya merupakan komponen web dari aplikasi J2EE secara keseluruhan. JSP juga memerlukan JVM (Java Virtual Machine) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal Java Virtual Machine di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut dengan Web Container.

Teknologi JSP menyediakan cara yang lebih mudah dan cepat untuk membuat halaman-halaman web yang menampilkan isi secara dinamik. Teknologi JSP didesain untuk membuat lebih mudah dan cepat dalam membuat aplikasi berbasis web yang bekerja dengan berbagai macam web server, application server, browser dan development tool.

#### **2.1.2 Daur Hidup JSP**

Sebagai gambaran bagaimana JSP melalui masa hidupnya bisa dilihat pada gambar berikut :



Gambar 2.1 Daur hidup JSP

Seperti tipe aplikasi java lainnya (Servlet, Applet, Midlet dll), JSP juga bertipe strong Type artinya penggunaan variable pada halaman tersebut harus dideklarasikan terlebih dahulu. Misalnya pada sintaks pengulangan berikut:

```

for (int i=1; i<13; i++)
{
    // statement
}

```

Seperti halnya skrip-skrip server side yang lain, JSP pun memerlukan Web server. Skrip ASP memerlukan IIS sebagai web server, PHP memerlukan IIS atau Apache, sedangkan JSP bisa menggunakan Apache Tomcat sebagai salah satu web server yang mendukungnya.

Agar bisa menjalankan file-file JSP yang berbasis Java, diperlukan web server yang mampu memproses Java, atau minimal JSP engine yang dapat terintegrasi dengan web server.

### 2.1.3 Konsep Pemrograman JSP

Web tidak hanya digunakan untuk menampilkan informasi, tetapi juga digunakan agar user dan server dapat berkomunikasi. Untuk kebutuhan ini, tidak mudah membuat web yang menarik dan informatif hanya menggunakan HTML, tetapi diperlukan request dari client side dan respon dari server side.

JSP adalah salah satu dari banyak bahasa pemrograman web yang berhubungan dengan suatu data (menambah, menghapus, mengubah dan menampilkan ke user). JSP juga dapat membuat suatu halaman web dapat berinteraksi langsung dengan user (dynamic web site), sehingga dengan JSP pembuatan halaman web tidak hanya dapat menampilkan data, tetapi juga berhubungan dengan bagaimana data tersebut digunakan dalam suatu kondisi tertentu. Selain itu, JSP dapat membuat suatu server untuk dapat mengembalikan request yang diminta user terhadap server tersebut.

Keuntungan dalam bahasa pemrograman web JSP adalah sebagai berikut:

- Multi platform* (dapat dijalankan pada lingkungan system Windows, UNIX, dan juga LINUX)
- Komponen *reuse* (memudahkan untuk mengembangkan dan menggunakan karena obyek-obyek dapat dikembangkan dengan mudah) (Rickyanto, 2002)

### 2.1.4 Elemen-elemen JSP

JSP merupakan bahasa pemrograman yang bersifat *embedded* pada kode HTML maupun WML, namun juga dapat berdiri sendiri dengan sintaks JSP.

JSP memiliki bermacam-macam elemen yang dapat digunakan dalam suatu halaman JSP. Elemen JSP meliputi:

#### a. Direktif

*Direktif* digunakan untuk menentukan nilai umum, seperti deklarasi kelas, *method* yang diimplementasikan, output *content type*, dan lain-lain. *Direktif* tidak menghasilkan output apapun untuk *client*. Seluruh *direktif* memiliki pengaruh terhadap keseluruhan kode yang terdapat pada *file* JSP, dan hanya *file* JSP. *Direktif* secara keseluruhan adalah sebagai berikut:

```

<%@ directiveName attribute="value"
attribute ="value" %>

```

*Direktif* terdiri atas tiga macam, yaitu:

#### ▪ Direktif Page

*Direktif page* digunakan untuk mendefinisikan atribut yang penting bagi keseluruhan kode dalam halaman tersebut. Sebuah JSP tunggal dapat mengandung banyak *direktif page*, dan melalui penterjemahan seluruh *direktif page* diterima dan digunakan bersama-sama pada halaman yang sama. *Direktif page* merupakan yang sangat penting, karena memiliki pengaruh terhadap keseluruhan kode yang terdapat pada *file* JSP yang mengandung *direktif* ini. Sintaks *direktif page* sebagai berikut:

```

<%@ page ATTRIBUTE %>

```

Tabel 2.1 Tabel Direktif page

Atribut	Deskripsi	Default
<i>Language</i>	Mendefinisikan bahasa <i>scriptlet</i> yang digunakan	Java
<i>import</i>	Untuk mengimport <i>package</i> atau kelas java sebagaimana program Java umumnya	
<i>session</i>	Menentukan apakah halaman JSP menggunakan HTTP <i>session</i> .	"true"
<i>Content Type</i>	Mendesripsikan <i>encoding</i> karakter yang digunakan serta tipe MIME sebagai <i>response</i> dari halaman JSP. Penggunaan: "TYPE=type" atau "TYPE=type MIME; charset=CHARSET", CHARSET adalah karakter <i>encoding</i> yang digunakan.	Default Type adalah "text/html" dan nilai default CHARSET adalah ISO-8859-1
<i>isThread Safe</i>	Mendefinisikan tingkat keamanan mengenai masalah <i>threading</i> dari halaman JSP. Jika "false", request akan diproses sebagai <i>Thread tunggal</i> , berurutan sesuai urutan datangnya request.	"true"
<i>buffer</i>	Menentukan model <i>buffering</i> untuk output <i>stream</i> ke	Default adalah

	<i>client</i> . Apabila nilainya diisi <i>none</i> , maka tidak terjadi <i>buffering</i> .	dilakukan <i>buffering</i> maksimal 8 kb.
<i>info</i>	Mendefinisikan string informasi yang dapat diperoleh dari implementasi metode <i>Servlet.getServletInfo()</i> .	
<i>isErrorPage</i>	Mengindikasikan apakah halaman JSP ini merupakan halaman <i>error</i> dari halaman JSP lain.	"false"
<i>ErrorPage</i>	Mendefinisikan <i>URL</i> dari <i>file</i> JSP lain yang dipanggil, apabila terjadi <i>error</i> saat <i>file</i> JSP diproses	
<i>autoFlush</i>	Apabila "true", output <i>buffer</i> akan di <del>flush</del> / dikeluarkan setelah <i>buffer</i> penuh/ <i>full</i> . Apabila "false", pesan <i>exception</i> akan dikeluarkan untuk mengindikasikan jika <i>buffer</i> telah penuh.	"true"
<i>extends</i>	Digunakan untuk men- <i>generate superclass</i>	

- **Direktif Include**  
*Direktif Include* digunakan untuk menyisipkan isi *file* lain ke dalam *file* JSP. Sintaks penulisan *direktif include* sebagai berikut:

```
<%@ include file="filename" %>
```

- **Direktif Taglib**  
*Direktif Taglib* mengijinkan penggunaan *tag* tambahan atau *library tag* di dalam halaman. Sintaks penulisan *direktif taglib* sebagai berikut:

```
<%@ taglib uri="taglibLibraryURI"
prefix="tagPrefix" %>
```

**Tabel 2.2 Tabel Direktif Taglib**

Atribut	Deskripsi	Default
<i>uri</i>	<i>Uniform Resource Identifier</i> (URI) memperkenalkan <i>tag library descriptor</i> digunakan untuk penamaan khusus sejumlah <i>tag</i> tambahan dan mengatakan kepada <i>container</i> apa yang harus dilakukan dengan <i>tag</i> khusus tersebut.	Tidak ada nilai yang disebabkan kesalahan kompilasi
<i>TagPrefix</i>	<u>Menetapkan prefix</u> dalam <code>&lt;prefix&gt;:&lt;tagname&gt;</code> yang digunakan untuk menetapkan <i>tag</i> tambahan. Catatan: prefix JSP, JSPx, java, javax, servlet, sun, dan sunw telah disediakan.	Tidak ada nilai yang disebabkan kesalahan kompilasi

## b. Scripting

Elemen *scripting* digunakan untuk memasukkan kode *scripting* di dalam JSP. Elemen *scripting* mengijinkan untuk mendeklarasikan variabel dan *method*, termasuk wewenang kode *scripting*, dan menilai ekspresi.

Ada tiga tipe elemen *scripting* yaitu:

- **Deklarasi**  
Deklarasi digunakan untuk mendeklarasikan variabel atau *method*. Contoh penulisan deklarasi sebagai berikut:

```
<%!
    Private int i=40;
    Public void myMethod()
    {
        //Program dimasukkan di
        sini
    }
%>
```

- **Ekspresi**  
Ekspresi digunakan untuk ekspresi dalam Java dan menampilkannya sebagai *string* pada *browser*. Contoh penggunaan ekspresi sebagai berikut:

```
<%! int i=0; %>
<%
    i++;
%>
Hello World ! <%= "This JSP has been
accessed "+I +"times" %>
```

- **Scriptlet**  
*Skriptlet* digunakan untuk deklarasi, ekspresi, dan kode lain. Contoh penggunaan *skriptlet* sebagai berikut:

```
<%
    For (int i=0;i<10;i++)
    {
        Out.println("<b>Hello World This
is a loop test "+I +"</b>");
        System.out.println("This goes to
the system.out stream "+i);
    }
%>
```

## c. Standart Action

*Standart action* adalah *tag* yang didefinisikan dalam spesifikasi JSP. *Standart action* meliputi:

- **JSP:useBean**  
*Tag* ini digunakan untuk mengasosiasikan *Java Bean* dalam JSP sehingga kita dapat menggunakan objek yang terdapat dalam *Java Bean* untuk diaplikasikan dalam kode JSP. Sintaks penulisan *JSP:useBean* sebagai berikut:

```
<JSP:useBean id="name"
scope="page|request|session|application"
beanDetails />
```

*Beandetails* adalah salah satu dari :

- *Class*="className"
- *Class*="className" *type*="typeName"
- *beanName*="beanName" *type*="typeName"
- *type*="typeName"

Contoh penggunaan *JSP:useBean* adalah sebagai berikut:

```
<JSP:useBean id="param" scope="session"
class="java.lang.String">
```

- **JSP:setProperty**  
Tag ini digunakan untuk melengkapi penggunaan tag `<JSP:useBean>`. Sintaks penulisan `JSP:setProperty` adalah sebagai berikut:

```
<JSP:setProperty name="beanName"
propertydetails/>
```

*Propertydetails* adalah salah satu dari:

- *Property*="\*"
- *Property*="propertyName"
- *Property*="propertyName" *param*="parameterName"
- *Property*="propertyName" *value*="propertyValue"

Contoh penggunaan `JSP:setProperty` adalah sebagai berikut:

```
<JSP:setProperty name="namaBean"
property="*" />
```

- **JSP:getProperty**  
Tag ini digunakan untuk mendapatkan nilai *property* dari *bean* dan merupakan pelengkap dari tag `<JSP:getProperty>`. Sintaks penulisan `JSP:getProperty` adalah sebagai berikut:

```
<JSP:getProperty name="name"
property="propertyName" />
```

- **JSP:param**  
Tag ini digunakan untuk melengkapi tag `<JSP:plugin>` maupun `<JSP:forward>`. *Translation error* akan terjadi apabila digunakan di tempat lain. Sintaks penulisan `<JSP:param>` adalah sebagai berikut:

```
<JSP:param name="paramname"
value="paramvalue" />
```

- **JSP:params**  
Tag ini digunakan hanya dalam `<JSP:plugin>` saja. Penggunaan di tempat lain akan mengakibatkan *translation time error*.

- **JSP:include**  
Tag ini digunakan untuk mengikutsertakan file lain dalam JSP saat *request* dilakukan. Dalam sebuah *action* `<JSP:include>` dapat terdapat satu atau lebih tag `<JSP:param>` di dalamnya. Sintaks penulisan `<JSP:include>` adalah sebagai berikut:

```
<JSP:include page="filename"
flush="true" />
```

Atau

```
<JSP:include page="urlSpec"
flush="true">
  <JSP:param name="paramname"
value="paramvalue" />
  ...
</JSP:include>
```

- **JSP:forward**  
Tag ini digunakan untuk mem-forward ke file JSP, Servlet, atau file *static* lain. Sintaks penulisan `<JSP:forward>` adalah sebagai berikut:

```
<JSP:forward page="url" />
```

Atau

```
<JSP:forward page="urlSpec" />
  <JSP:param name="paramname"
value="paramvalue" />
  ...
</JSP:forward>
```

- **JSP:plugin**  
Tag ini digunakan untuk menghasilkan kode-kode yang sesuai target *browser* dalam menyisipkan komponen Java Bean atau Applet secara otomatis. `JSP:plugin` dapat memiliki dua tag pendukung tambahan yaitu:
  - Elemen `<JSP:param>`, untuk menempatkan parameter tambahan ke dalam komponen *Applet* atau *Java Bean*.
  - Elemen `<JSP:fallback>`, untuk menentukan isi yang akan ditampilkan pada *browser client* jika *plugin* tidak dapat dijalankan.

- **JSP:fallback**  
Tag ini digunakan sebagai *sub* elemen dari `<JSP:plugin>` yang berfungsi untuk menampilkan pesan, apabila proses *plugin* mengalami *error*. Contoh penggunaan `JSP:plugin` dan `JSP:fallback` adalah sebagai berikut:

```
<JSP:plugin type="bean|applet"
code="objectCode"
codebase="objectCodebase"
align="alignment" archive="archiveList"
height="height" hspace="hspace"
jreversion='jreversion'
name="componentName" vspace="vspace"
width="width" nspluginurl="url"
iepluginurl="url">
  <JSP:params>
    <JSP:param name="paramname"
value="paramvalue" />
    <JSP:param name="paramname"
value="paramvalue" />
    ...
  </JSP:params>
  <JSP:fallback> Alternate Text to
Display </JSP:fallback>
</JSP:plugin>
```

## 2.1.5 JSP dan Database

Dalam pemrograman *web* yang bersifat *server side*, harus dilakukan pemrosesan data terhadap suatu basis data. Perintah-perintah yang digunakan dalam JSP untuk melakukan proses dengan suatu basis data adalah sebagai berikut:

### a. Connection

Digunakan untuk membuat suatu koneksi terhadap suatu basis data tertentu agar dapat diakses. Bentuk dari penggunaan *connection* adalah sebagai berikut:

```
<%
String user="root";
String pass='123456'
String url="jdbc:odbc:TA";
Connection
con=DriverManager.getConnection(url,pass)
%>
```

#### b. Statement

Digunakan agar dapat membuat suatu perintah SQL yang akan dijalankan atau diproses. Bentuk dari penggunaan *statement* adalah sebagai berikut:

```
<%
Statement st=con.createStatement();
%>
```

#### c. Querying

Digunakan untuk menjalankan perintah SQL yang mengambil data dari suatu basis data. Bentuk dari penggunaan *querying* adalah sebagai berikut:

```
<%
String Query="SELECT * FROM TA";
ResultSet rs=st.executeQuery(query);
%>
```

#### d. Memproses Hasil

Digunakan untuk memproses hasil sesuai *query* yang disimpan di dalam objek *resultSet*. Bentuk memproses hasil sebagai berikut:

```
<%
While(rs.next()){
    out.println(rs.getString(1)+"
"+rs.getString(2));
}
%>
```

#### e. Menutup Koneksi

Sebelum menutup koneksi *database*, objek *resultSet* harus dilepas dengan kode sebagai berikut:

```
Statement.close();
```

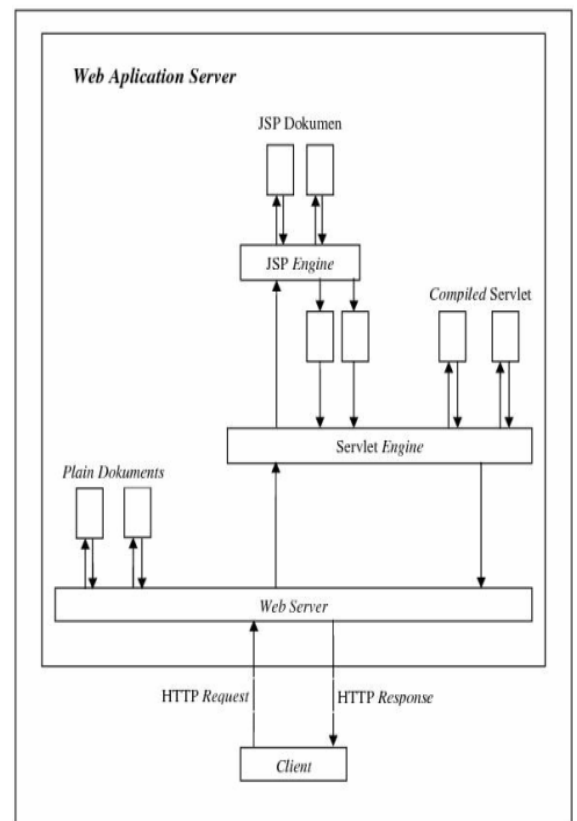
Untuk menutup koneksi *database* dapat dilakukan dengan kode sebagai berikut:

```
connection.close();
```

### 2.1.6 Arsitektur Web Aplikasi Server

Saat pertama kali *web* aplikasi *server* menerima *request* untuk halaman JSP, maka mesin JSP akan *meng-compile* halaman sumber ke Servlet. Dasarnya, halaman JSP yang mengandung HTML atau halaman XML dengan perintah JSP di dalamnya akan dikembalikan menjadi kelas Servlet dengan *method* yang mengandung perintah untuk mencetak halaman dan perintah dari sumber JSP. Proses ini juga terjadi saat halaman JSP mengalami perubahan. Servlet sementara dijalankan oleh mesin Servlet untuk membuat pesan jawaban yang akan dikirimkan ke *client*.

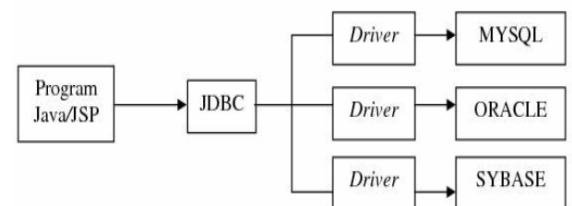
Gambar dibawah ini menunjukkan *web* aplikasi *server* yang terbagi dalam tiga *layer* dasar yaitu *web server*, mesin Servlet, dan mesin JSP.



Gambar 2.2 Arsitektur Web Application Server

### 2.2 Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC) adalah Application Programming Interface (API) yang dikembangkan oleh Sun Microsystems untuk menyediakan akses data *universal* dalam bahasa pemrograman Java. JDBC merupakan bagian dari Java 2 SDK Standart Edition. Inti JDBC adalah *package java.sql* dan *javax.sql* (bagian dari J2EE). JDBC menyediakan mekanisme untuk koneksi *database*, sintaks untuk mengirimkan *query* dan melakukan transaksi serta struktur data yang mewakili hasil *query*.



Gambar 2.3 Struktur JDBC

JDBC merupakan koleksi API yang terdiri dari sekumpulan *class* dan *interface* yang ditulis dalam pemrograman Java sebagai standart API untuk *developer database* maupun *software database* sehingga memungkinkan pembuatan aplikasi *database* yang *portable* dengan bahasa pemrograman Java. (Rickyanto, 2002)

Dengan JDBC kita dapat membuat program aplikasi *database* yang dapat berganti *database server* tanpa mengubah banyak kode dari program yang sudah ada sehingga memberikan portabilitas tinggi bagi aplikasi yang telah dibuat.

Dalam JDBC terdapat tujuh langkah standar untuk menangani *database*:

- Memanggil / *load driver* JDBC
- Mendefinisikan koneksi *database*
- Melakukan koneksi *database*
- Membuat obyek *statement*
- Melakukan *query* atau *update*
- Memproses hasil
- Menutup koneksi

## 2.3 Entity Relationship Diagram (ERD)

### 2.3.1 Definisi ERD

Entity Relation Diagram (ERD) adalah pemodelan yang menggambarkan data sebagai sekumpulan entity, atribut, dan relationship. ERD merupakan dokumen data perusahaan dalam ringkasan cara dengan mengidentifikasi tipe data entity dan hubungan antar entity. Atribut merupakan properties atau bagian dari suatu entity. Relationship menggambarkan hubungan antar entity.

### 2.3.2 Tipe Entity

Tipe entity dapat berupa elemen lingkungan, sumber, atau transaksi yang penting bagi perusahaan untuk didokumentasikan dengan data. Contoh tipe entity adalah customer, karyawan, dan time card. Tipe entity didokumentasikan dalam ERD dengan bentuk persegi. Setiap persegi diberi nama berdasarkan nama tipe entity, biasanya digunakan kata benda. Pemodelan entity tampak dalam gambar 2.4



Gambar 2.4 Entity

### 2.3.3 Relasi Dalam ERD

Relasi adalah asosiasi yang ada antara dua tipe entity. Relasi digambarkan dalam bentuk belah ketupat. Setiap belah ketupat diberi nama menggunakan kata kerja. Seperti contoh pada gambar 2.4., setiap karyawan mengisi time card. Relasi ini dapat juga dibaca sebaliknya yaitu time card diisi oleh karyawan. Pemodelan relasi dalam ERD tampak dalam gambar 2.5



Gambar 2.5 relasi dalam ERD

### 2.3.4 Konektivitas

Jumlah banyaknya suatu entity dapat berelasi dengan entity yang lain dikenal dengan istilah konektivitas. Ada tiga jenis konektivitas yang digunakan untuk menggambarkan hubungan antar entity, yaitu:

#### a. One To One

Suatu hubungan entity di mana item suatu entity tepat memiliki hubungan dengan satu item pada

entity lain. Diagram One To One tampak dalam gambar 2.6



Gambar 2.6 Diagram One To One

#### b. One To Many

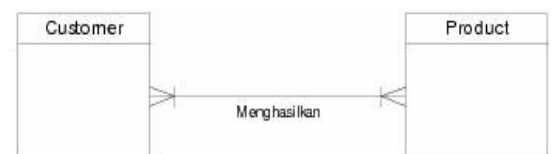
Suatu hubungan entity di mana item pada suatu entity boleh mempunyai hubungan dengan dua atau lebih item pada entity lain. Diagram One To Many tampak dalam gambar 2.7



Gambar 2.7 Diagram One To Many

#### c. Many To Many

Suatu hubungan entity di mana item pada suatu entity dapat mempunyai hubungan dengan lebih dari satu item pada entity lain, demikian pula sebaliknya. Diagram Many To Many tampak dalam gambar 2.8



Gambar 2.8 Diagram Many To Many

### 2.3.5 Mandatory

Fungsi mandatory melambangkan bahwa entity harus mempunyai item terlebih dahulu, baru item yang lain bisa diisi item dan dihubungkan. Tipe garisnya hampir sama, hanya saja tanda 'o' diganti dengan '|'. Gambar pemodelan tipe garis dengan mandatory tampak pada gambar 2.9

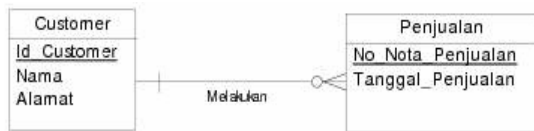


Gambar 2.9 Tipe Garis Dengan Mandatory

### 2.3.6 Atribut

Atribut merupakan karakteristik entity. Contoh: atribut entity customer adalah kode customer, nama customer, alamat, telepon. Sebenarnya, atribut merupakan elemen data, dan setiap atribut diberi nilai tunggal, yang disebut nilai atribut. Atribut yang memperkenalkan entity disebut identifiers. Contoh identifiers adalah kode customer. Tidak ada dua customer yang memiliki kode yang sama. Atribut lain yang menjelaskan entity disebut descriptors. Contoh descriptors adalah nama customer, alamat, dan telepon. Identifiers ditunjukkan dalam ERD dengan memberi garis bawah pada atribut identifiers entity

tersebut. Gambar identifiers dalam ERD tampak dalam gambar 2.10



**Gambar 2.10 Identifiers Dalam ERD**

## 2.4 Entity Relational Diagram (ERD)

### 2.4.1 Definisi DFD

Data Flow Diagram (DFD) adalah sebuah gambar yang mewakili sistem menggunakan sejumlah bentuk simbol untuk menggambarkan bagaimana aliran data melalui proses yang saling berhubungan. (McLeod, 1998)

### 2.4.2 Simbol-Simbol Dalam DFD

Ada empat simbol yang digunakan untuk menggambarkan aliran data dalam sebuah sistem. Simbol-simbol tersebut digunakan untuk mewakili elemen lingkungan (external entity), proses, aliran data, dan data storage.

#### 2.4.2.1 Elemen Lingkungan (External Entity)

Elemen lingkungan berada di luar batas sistem. Elemen ini menyediakan sistem suatu input data dan menerima output data. Dalam DFD tidak ada perbedaan yang terbentuk antara data dan informasi. Semua aliran dianggap sebagai data. Nama terminator digunakan untuk menggambarkan elemen lingkungan. Terminator digambarkan dalam DFD dalam bentuk persegi. Setiap simbol terminator diberi nama sesuai nama elemen lingkungan. Simbol elemen lingkungan (external entity) tampak dalam gambar 2.11



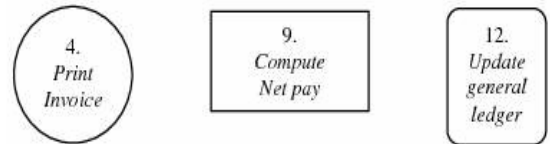
**Gambar 2.11 Simbol Elemen lingkungan**

Suatu terminator dapat berupa:

- Orang, seperti manager, yaitu orang yang menerima report dari sistem
- Organisasi, seperti departemen lain dalam perusahaan atau perusahaan lain.
- Sistem lain

#### 2.4.2.2 Proses

Proses adalah sesuatu yang mengubah input menjadi output. Proses digambarkan dalam bentuk lingkaran, persegi horizontal, atau persegi vertical dengan sisi yang dibulatkan. Tiap simbol proses diberi nama. Teknik pemberian nama yang paling sering menggunakan kata kerja dan sebuah obyek, tetapi dapat juga digunakan nama sistem atau program komputer. Simbol proses tampak dalam gambar 2.12



**Gambar 2.12 Simbol Proses**

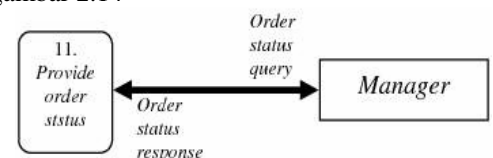
### 2.4.2.3 Aliran Data

Aliran data terdiri dari kumpulan relasi logical elemen data yang dikirimkan dari satu titik atau proses lainnya. Simbol panah digunakan untuk menggambarkan aliran. Jumlah data diwakili dengan aliran data tunggal, dapat berasal dari bermacam-macam elemen data tunggal menuju satu atau banyak file. Simbol aliran data tampak dalam gambar 2.13



**Gambar 2.13 Simbol Aliran Data**

Aliran data terdiri dari satu atau lebih structure data. Structure adalah kelompok elemen data yang menggambarkan sebagian item atau transaksi. Lebih mudah untuk membayangkan structure sebagai susunan elemen data yang menyusun record, atau sebagai kumpulan relasi elemen yang dicetak pada dokumen. Aliran data dapat bercabang ketika data yang sama dikirimkan ke banyak lokasi dalam sistem. Kadang kala desain sistem akan memanggil aliran dua arah. Simbol aliran data dua arah tampak dalam gambar 2.14



**Gambar 2.14 Simbol Aliran Dua Arah**

### 2.4.2.4 Data Storage

Saat diperlukan untuk menyimpan data untuk alasan tertentu, data store digunakan. Dalam DFD, data store adalah tempat penyimpanan data. Ada beberapa pilihan penggambaran data store dengan kumpulan garis parallel, persegi open ended, atau oval. Simbol data store tampak dalam gambar 2.15



**Gambar 2.15 Simbol Data Store**



## 2.5 Hypertext Markup Language (HTML)

### 2.5.1 Definisi HTML

Hypertext Markup Language (HTML) digunakan untuk mempersiapkan sebuah dokumen hypertext. HTML sebenarnya bukan bahasa pemrograman, karena tercermin dari namanya. HTML adalah suatu bahasa mark up. HTML digunakan untuk melakukan mark up (penandaan) terhadap sebuah dokumen teks. (Graham, 1995).

Simbol mark up yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Kedua tanda ini disebut tag. Tag yang digunakan sebagai tanda penutup diberi karakter garis miring (</...>). Berikut contoh penggunaan tag HTML:

```
<H> Ini adalah heading</H>
```

### 2.5.2 Struktur Dasar HTML

Sebuah file HTML merupakan file teks biasa yang mendukung tag-tag HTML. Karena merupakan file teks, maka HTML dapat dibuat dengan teks editor yang sederhana, misalnya NotePad.

Secara lengkap, file HTML mempunyai bagian head dan bagian body. Struktur lengkap HTML adalah sebagai berikut:

```
<html>
<head>
.....
</head>
<body>
.....
</body>
</html>
```

Tag HTML tidak bersifat case sensitive, jadi <HTML> akan sama dengan <html>.

### 2.5.3 Bagian Head

Bagian head tidak harus ada pada dokumen HTML, tetapi pemakaian head yang benar akan meningkatkan kegunaan suatu dokumen HTML. Bagian head mengandung informasi mengenai dokumen. Isi bagian head, kecuali judul dokumen, tidak akan terbaca oleh user.

Elemen yang umum ditemukan pada bagian head adalah tag <title> kemudian ditutup dengan </title>.

### 2.5.4 Bagian Body

Bagian body merupakan isi dari dokumen HTML. Semua informasi yang akan ditampilkan, mulai dari teks, gambar, sound, dan lain-lain akan ditempatkan di bagian ini. Kita mengawali bagian body dengan tag <body> dan mengakhirinya dengan tag </body>. Beberapa atribut dapat ditambahkan pada bagian body, dan berfungsi untuk:

- bgcolor untuk menentukan warna background dokumen
- link untuk menentukan warna link
- text untuk menentukan warna teks, dan lain-lain.

Berikut ini adalah contoh kode HTML yang menggunakan atribut-atribut body:

```
<html>
<head>
<title>contoh bagian body</title>
```

```
</head>
<body
      bgcolor="#339900"
      text="#ffffff" link="0000ff">
</body>
</html>
```

## 2.6 MySQL

### 2.6.1 Definisi MySQL

MySQL adalah suatu software sistem manajemen database. Database adalah suatu koleksi data yang terstruktur. MySQL merupakan Relational Database Management system (RDBMS) yang didistribusikan secara gratis di bawah lisensi General Public License (GPL). Sistem manajemen database seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di komputer.

MySQL menggunakan Standart Structured Query Language (SQL), yaitu bahasa standart yang paling banyak digunakan untuk mengakses database. Alasan penggunaan MySQL sebagai software database server adalah MySQL dari awal didesain untuk menangani database yang cukup sangat besar, lebih cepat dari solusi yang ada. Konektivitas, kecepatan dan security yang baik membuat MySQL sangat cocok digunakan di internet. (Rickyanto, 2002)

### 2.6.2 Keistimewaan MySQL

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL, yaitu:

#### a. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi diantaranya adalah windows, linux, freebsd, Mac OS X server, solaris, amiga, HP-UX, dan masih banyak lagi.

#### b. Open source

MySQL didistribusikan secara open source (gratis), di bawah lisensi GPL, sehingga dapat dipergunakan secara cuma-cuma tanpa dipungut biaya.

#### c. Multi user

MySQL dapat digunakan oleh beberapa user dalam waktu bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah database server mySQL dapat diakses client secara bersamaan.

#### d. Performance tuning

MySQL mempunyai kecepatan yang mengagumkan dalam menangani query sederhana.

#### e. Column types

MySQL memiliki tipe kolom yang sangat kompleks.

#### f. Command dan function

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah select dan where dalam query.

#### g. Security

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan ijin

akses user dengan sistem perijinan yang mendetail serta password terenkripsi.

*h. Scalability dan limits*

MySQL mampu menangani database dalam skala besar.

*i. Connectivity*

MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix Socket (UNIX), atau named pipes (NT).

*j. Localization*

MySQL dapat mendeteksi pesan kesalahan (error code) pada client menggunakan lebih dari dua puluhan bahasa.

*k. Interface*

MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman menggunakan fungsi API.

*l. Clients dan tools*

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

*m. Struktur table*

MySQL memiliki struktur table yang lebih fleksibel dalam menangani alter table, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle. (Prasetyo, 2003)

## 2.7 Tomcat

Tomcat merupakan sebutan untuk web container, yang merupakan implementasi resmi spesifikasi JSP dan servlet dari sun microsystem. Tomcat adalah program web server berbasis java untuk aplikasi servlet dan java server pages (JSP). Tomcat diperlukan dalam pemrograman JSP, karena JSP akan dijalankan pada server Tomcat.

Proyek Tomcat diawali oleh Jakarta project yang dipelopori oleh komunitas pembuat web server gratis dan open-source, apache. Mulanya Tomcat ini bernama ApacheJServ. Tomcat yang digunakan adalah Tomcat versi 4.1.24 dan dijalankan secara stand alone. Stand alone berarti Tomcat tidak dijalankan bersama web server, misalnya apache atau Microsoft IIS. Versi 4 dari Tomcat juga mendukung implementasi spesifikasi servlet 2.2 dan JSP 1.1 tanpa perubahan sehingga program JSP yang telah dibuat berdasarkan spesifikasi tersebut tidak perlu ditulis ulang atau disesuaikan dengan spesifikasi baru. (Rickyanto, 2002)

Di dalam direktori c:\Tomcat atau Tomcat\_Home terdapat subdirektori-subdirektori berikut:

- Bin, berisi skrip untuk menjalankan dan menghentikan program Tomcat.
- Conf, berisi bermacam-macam file konfigurasi termasuk server.xml (yaitu file konfigurasi utama dari Tomcat) dan web. XML yang mengeset nilai default untuk berbagai aplikasi web yang dijalankan di Tomcat.
- Doc, berisi dokumentasi dari Tomcat.
- Lib, berisi file-file pustaka (library) berupa file jar.
- Src, berisi file-file source dari api servlet webapps berisi contoh aplikasi web.

## 2.8 Macromedia Dreamweaver MX

### 2.8.1 Pengenalan Macromedia Dreamweaver MX

Macromedia Dreamweaver MX adalah program aplikasi professional untuk mengedit HTML secara visual dan mengelola web site serta pages. Program ini menyediakan banyak perangkat yang dapat meningkatkan kemampuan user di dalam membuat web. Program aplikasi Macromedia Dreamweaver MX menyertakan banyak perangkat yang berkaitan dengan pengkodean dan fitur seperti HTML, CSS, java script reference, dan java script debugger. Selain itu, program aplikasi ini juga meningkatkan pengeditan java script, XML, dan dokumen teks lainnya secara langsung, yaitu dengan menggunakan fasilitas code editors. (Agung, 2002)

Fitur-fitur pengeditan yang ditampilkan secara visual oleh Macromedia Dreamweaver MX dapat mempercepat penambahan desain dan fungsi pada halaman web tanpa harus menambahkan satu baris kode. Semua elemen yang terdapat dalam site dapat ditampilkan dan di-drag dari panel-panel (yang terdapat dalam Macromedia Dreamweaver MX) ke dalam dokumen secara langsung dan cepat.

Program aplikasi Macromedia Dreamweaver MX dapat mengimpor image atau movie yang dibuat dengan program aplikasi lainnya, seperti macromedia flash, macromedia fireworks, ataupun macromedia director. (Agung, 2002)

### 2.8.2 Insert Bar Dreamweaver MX yang Digunakan

Insert bar berisi tombol-tombol yang berfungsi membuat dan menyisipkan objek ke dalam dokumen. Secara default, insert bar memiliki 12 kategori, yaitu common, layout, text, table, frame, form, template, character, media, head, script, dan application.

Beberapa tombol-tombol pada insert bar, antara lain:

*a. Common*

Tab common berisi tombol-tombol untuk membuat dan menyisipkan objek yang paling sering atau biasa digunakan dalam dokumen. Tab common tersebut antara lain:

- Hyperlink, digunakan untuk memberikan link pada objek yang dipilih.
- Image, digunakan untuk menyisipkan image pada titik insertasi tempat kursor berada.
- Flash, digunakan untuk menempatkan movie pada titik insertasi.
- Navigation bar, digunakan untuk menyisipkan satu set image yang berfungsi sebagai tombol navigasi di dalam site.

*b. Layout*

Tab layout berisi tombol-tombol untuk membuat layout halaman. Tab layout tersebut antara lain:

- Table, digunakan untuk menyisipkan tabel pada titik insertasi.
- Draw layer, digunakan untuk membuat layer.

*c. Text*

Tab text berisi tombol-tombol yang berfungsi untuk melakukan editing text (misalnya cetak

huruf tebal, cetak miring) dan memformat paragraph (memilih heading).

#### d. Form

Tab form berisikan tombol-tombol untuk membuat form dan menyisipkan elemen form. Tab form tersebut antara lain:

- Forms, digunakan untuk menempatkan form pada titik insertasi.
- Text fields, digunakan untuk menempatkan kotak teks pada titik insertasi.
- Hidden field, digunakan untuk menempatkan kotak yang tersembunyi pada titik insertasi.
- Check box, digunakan untuk menempatkan check box pada titik insertasi.
- Radio group, digunakan untuk menempatkan tombol radio pada titik insertasi.
- List/menu, digunakan untuk menempatkan list atau menu pop-up pada titik insertasi.
- File field, digunakan untuk menempatkan kotak file pada titik insertasi.
- Button, digunakan untuk menempatkan tombol. Secara default, nama tombol akan diisi teks 'submit'.
- Label, digunakan untuk menyediakan cara yang structural berasosiasi dengan label teks.

#### e. Script

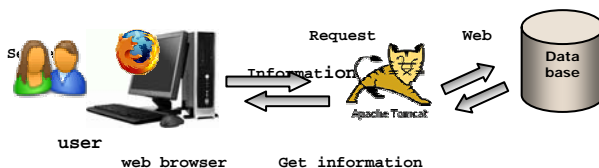
Tab script berisi tombol-tombol untuk menyisipkan elemen-elemen yang berkaitan dengan skrip ke dalam halaman. Script digunakan untuk menempatkan script pada titik insertasi.

### III PERANCANGAN DAN PEMBUATAN SISTEM

#### 3.1 DESAIN SISTEM

##### 3.1.1 Blok Diagram Sistem

Pada proyek akhir ini, blok diagram sistem dapat digambarkan sebagai berikut:

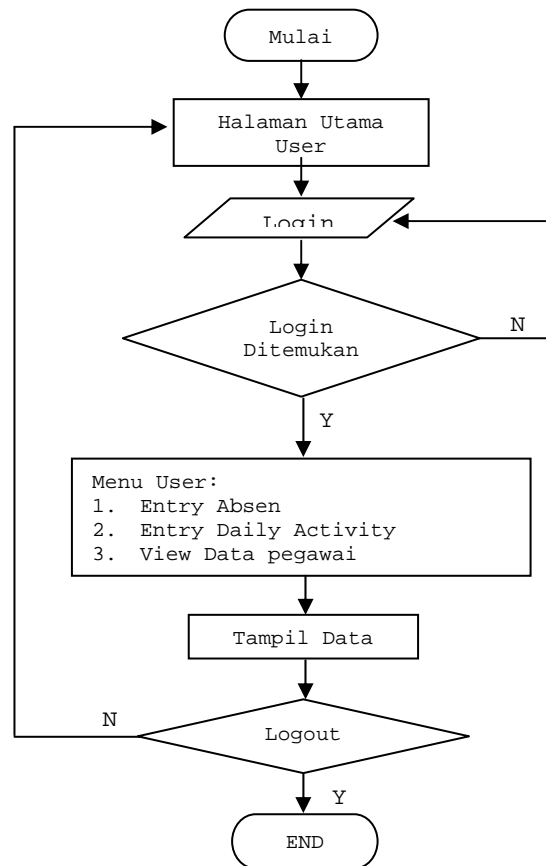


**Gambar 3.1 Desain sistem**

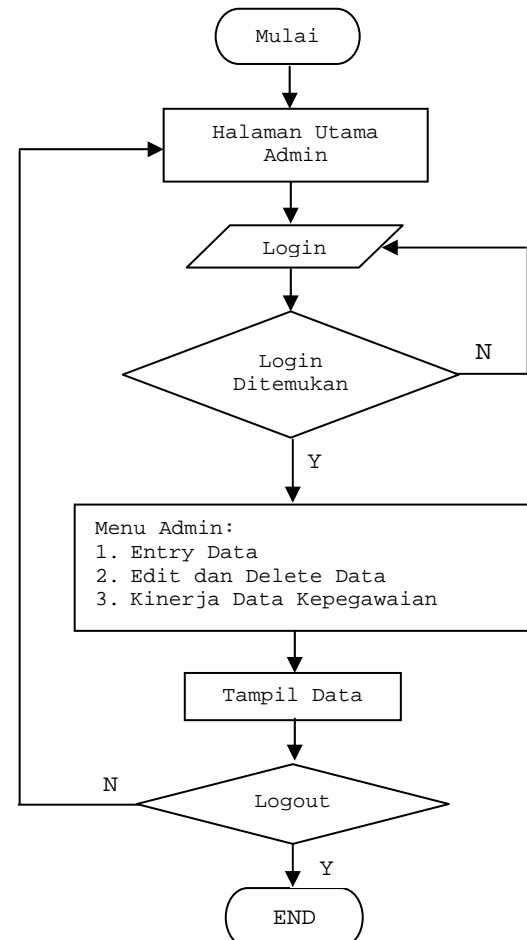
Dari desain sistem di atas terlihat bahwa user melalui web browser meminta informasi yang terdapat pada website analisa kinerja kepegawaian, web server akan melayani permintaan tersebut dengan mengambil data-data dari database yang dibutuhkan dan akan mengirimkan balasan berupa informasi yang diminta.

##### 3.1.2 Flowchart Sistem

Berikut adalah flowchart perancangan sistem dari proyek akhir sistem pembuatan analisa kinerja kepegawaian PENS-ITS menggunakan JSP :



**Gambar 3.2 Flowchart Sistem User**



**Gambar 3.3 Flowchart Sistem Admin**

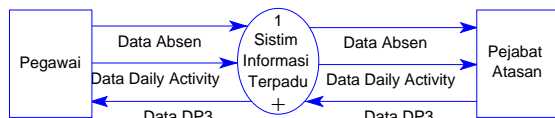
### 3.1.3 DFD (Data Flow Diagram)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Data Flow Diagram dalam pembuatan proyek akhir ini adalah sebagai berikut :

Pada DFD level 0 Sistem informasi terpadu didapat dari data absen dan data daily activity pegawai, kemudian sistem informasi terpadu mengirim data absensi dan daily activity kepada pejabat atasan. Data DP3 diperoleh dari pejabat atasan dan dikirimkan kepada sistem informasi terpadu untuk dikirimkan kepada pegawai.

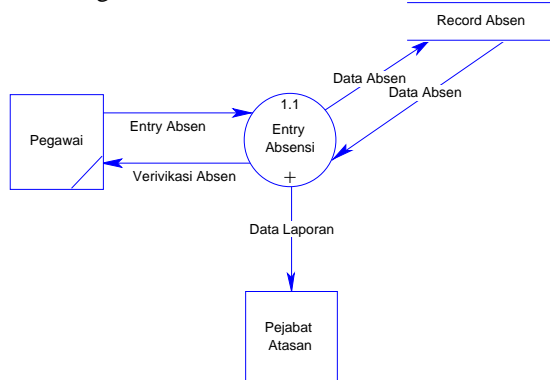
Berikut gambar DFD level 0 :



**Gambar 3.4 DFD level 0**

Pada DFD Level 1 Pegawai mengisi absen kemudian disimpan direcordstore absen, kemudian absen ditampilkan, diverivikasi, kemudian dilaporkan kepada pejabat atasan.

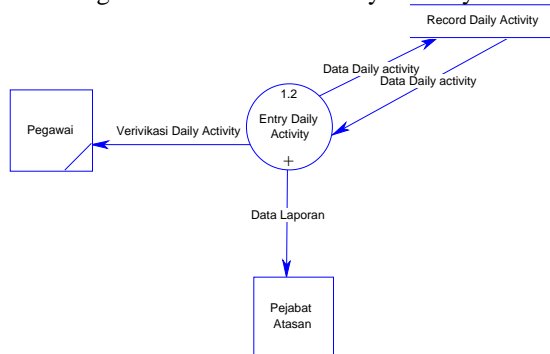
Berikut gambar DFD level 1 Absen :



**Gambar 3.5 DFD level 1 Absen**

Pegawai mengisi daily activity kemudian disimpan direcordstore daily activity, kemudian daily activity ditampilkan, diverivikasi, kemudian dilaporkan kepada pejabat atasan.

Berikut gambar DFD level 1 Daily Activity :

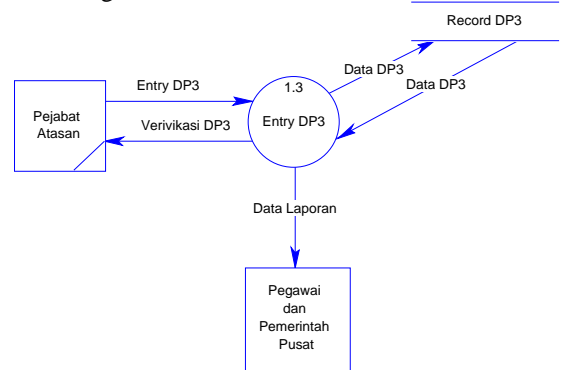


**Gambar 3.6 DFD level 1 Daily Activity**

Pejabat meng-entry DP3 kemudian disimpan direcordstore DP3, kemudian DP3 ditampilkan,

diverivikasi, dan kemudian dilaporkan kepada pegawai dan pemerintah pusat.

Berikut gambar DFD level 1 DP3 :

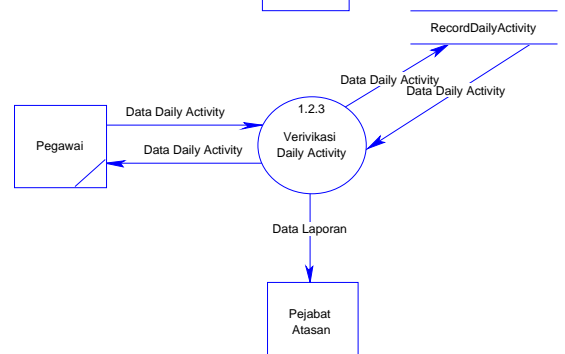
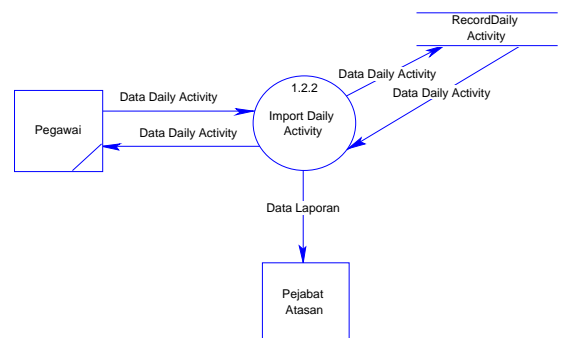
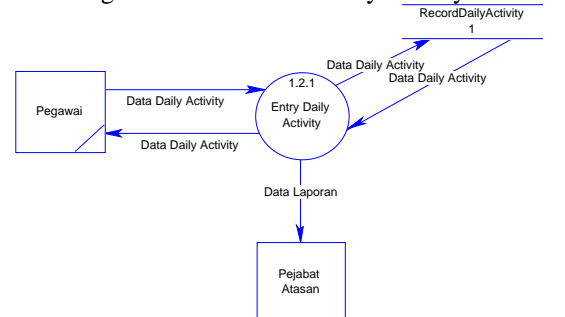


**Gambar 3.7 DFD level 1 DP3**

Pada DFD Level 2 Pegawai mengisi daily activity kemudian disimpan direcordstore daily activity, kemudian daily activity ditampilkan, diverivikasi, kemudian dilaporkan kepada pejabat atasan.

Pada DFD level 2 daily activity dibagi kedalam entry daily activity, import daily activity dan verivikasi daily activity yang nantinya akan dilaporkan kepada pejabat atasan.

Berikut gambar DFD level 2 daily activity :



**Gambar 3.8 DFD Level 2 Daily Activity**

### 3.1.4 ERD (Entity Relationship Diagram)

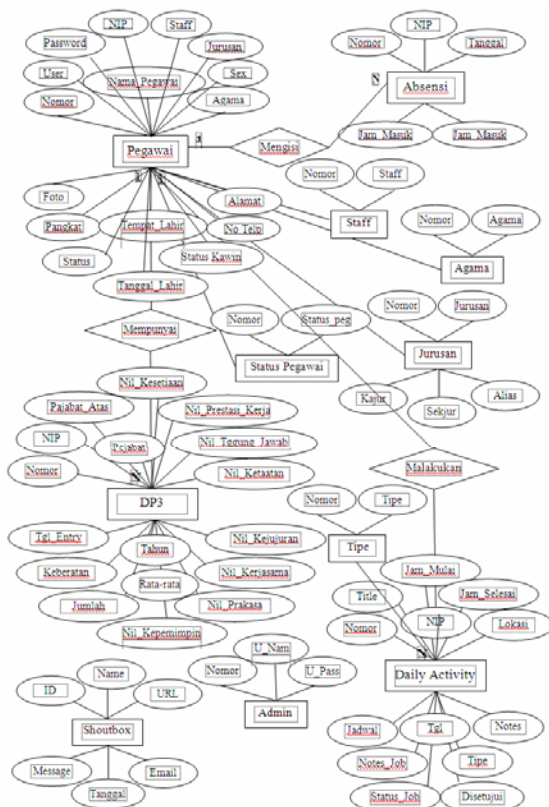
ERD adalah diagram yang memperlihatkan entitas-entitas yang terlibat dalam suatu sistem serta hubungan-hubungan relasi antar entitas tersebut.

Entity-Relationship merupakan suatu model perancangan data base yang sering digunakan. Entity bisa diartikan sebagai entitas sedangkan Relationship sebagai relasi, kedua komponen ini di deskripsikan lebih jauh melalui sebuah atribut tau propertis. Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya). Dan dapat dibedakan dari sesuatu yang lain. Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.

Pada model entity-relationship, semesta yang ada di dunia nyata di terjemahkan dengan memanfaatkan sejumlah perangkat konseptual menjadi sebuah diagram data, yang umumnya disebut dengan E-R Diagram.

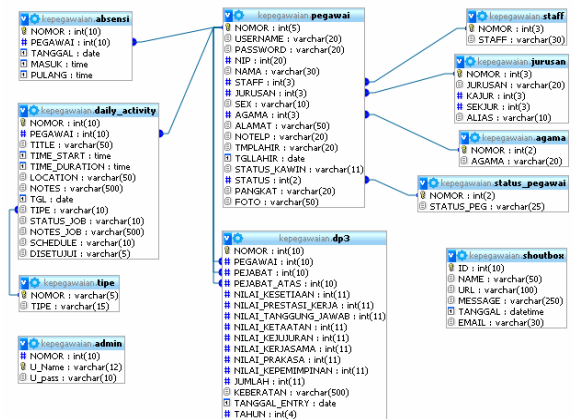
Entity Relationship Diagram (ERD) dalam pembuatan proyek akhir ini terlihat pada gambar dibawah ini :

Berikut ERD Konseptual :



Gambar 3.9 ERD Konseptual

Berikut Diagram ERD :



Gambar 3.10 Diagram ERD

### 3.1.5 Desain Tabel

Dari semua data diatas maka penyusunan tabel dari software pembuatan analisa kinerja kepegawaian PENS-ITS adalah :

Tabel 3.1 Tabel Admin

No.	Tabel User	Type	Keterangan
1.	Nomor	Int (5)	Nomor admin
2.	<u>U_Name</u>	Varchar (15)	Primary key User name admin
3.	U_Pass	Varchar (15)	Password admin

Tabel 3.2 Tabel Pegawai

No.	Tabel User	Type	Keterangan
1.	<u>Nomor</u>	Int (5)	Primary key Nomor urut pegawai
2.	Username	Varchar (20)	User Pegawai
3.	Password	Varchar (20)	Password Pegawai
4.	NIP	Int (20)	NIP Pegawai
5.	Nama	Int (20)	Nama Pegawai
6.	<u>Staff</u>	Int (3)	Foreign key Staff pegawai
7.	<u>Jurusan</u>	Int (3)	Foreign key Jurusan pegawai
8.	Sex	Varchar (10)	Jenis kelamin
9.	<u>Agama</u>	Varchar (10)	Foreign key Agama
10.	Alamat	Varchar (50)	Alamat
11.	Notelp	Int (15)	No telephone
12.	Tmp_lahir	Varchar (20)	Tempat lahir
13.	Tgl_lahir	Date	Tanggal lahir
14.	Status_Kawin	Varchar (10)	Status kawin
15.	<u>Status</u>	Int (2)	Foreign key

			Status pegawai
16.	Pangkat	Varchar (20)	Pangkat Pegawai

**Tabel 3.3 Tabel Absen**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Primary key Nomor urut
2.	<u>Pegawai</u>	Int (20)	Foreign key Nomor pegawai
3.	Tanggal	Date	Tanggal masuk
4.	Masuk	Time	Jam masuk
5.	Pulang	Time	Jam pulang

**Tabel 3.4 Tabel Daily Activity**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (10)	Primary key Nomor urut pegawai sebagai primary key
2.	<u>Pegawai</u>	Int (10)	Foreign key Nomor pegawai
3.	Title	Varchar (50)	Nama aktifitas
4.	Time_start	Time	Jam mulai
5.	Time_duration	Time	Jam selesai
6.	Location	Varchar (50)	Lokasi daily activity
7.	Notes	Varchar (500)	Catatan daily activity
8.	Tgl	Date	Tanggal entry
9.	<u>Tipe</u>	Int (2)	Foreign key Tipe
10.	Status job	Int (2)	Status
11.	Notes job	Varchar (500)	Catatan daily aktifitas
12.	Schedule	Int (10)	Jadwal
13.	Disetujui	Int (2)	Disetujui Atasan

**Tabel 3.5 Tabel DP3**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Primary key Nomor urut
2.	<u>Pegawai</u>	Int (5)	Foreign key Nomor urut pegawai
3.	<u>Pejabat</u>	Int (5)	Foreign key Pejabat
4.	<u>Pejabat atas</u>	Int (5)	Foreign key Pejabat atas
5.	Nilai_Kesetiaan	Int (8.5)	Nilai kesetiaan
6.	Nilai_pres_kerja	Int (8.5)	Nilai prestasi kerja
7.	Nilai_tanggung_	Int (8.5)	Nilai tanggung

	jawab		jawab
8.	Nilai_ketaatan	Int (8.5)	Nilai ketaatan
9.	Nilai_kejujuran	Int (8.5)	Nilai kejujuran
10.	Nilai_kerjasama	Int (8.5)	Nilai kerjasam
11.	Niali_prakasa	Int (8.5)	Nilai prakasa
12.	Nilai_kepemimpi-Nan	Int (8.5)	Nilai kepemimpinan
13.	Jumlah	Int (8.8)	Jumlah nilai
14.	Keberatan	Varchar (250)	Keberatan
15.	Tanggal_entry	Date	Tanggal entry
16.	Tahun	Int (4)	Tahun

**Tabel 3.6 Tabel Staff**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Primary key Nomor staff
2.	Staff	Varchar (30)	Nama staff

**Tabel 3.7 Tabel Jurusan**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Primary key Nomor jurusan
2.	Jurusan	Varchar (20)	Nama jurusan
3.	Kajur	Int (3)	Kajur
4.	Sekjur	Int (3)	Sekjur
5.	Alias	Varchar (10)	Nama Alias

**Tabel 3.8 Tabel Agama**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Primary key Nomor Agama
2.	Agama	Varchar (20)	Nama agama

**Tabel 3.9 Tabel Status Pegawai**

No.	Tabel User	Type	Keterangan
1.	<u><b>Nomor</b></u>	Int (5)	Nomor status
2.	Status_Pegawai	Varchar (25)	Status pegawai

**Tabel 3.10 Tabel Tipe**

No.	Tabel User	Type	Keterangan
1.	Nomor	Int (5)	Primary key Nomor tipe
2.	Tipe	Varchar (15)	Tipe daily activity

**Tabel 3.11 Tabel Shoutbox**

No.	Tabel User	Type	Keterangan
1.	<u><b>ID</b></u>	Int (10)	Primary key ID
2.	Name	Varchar (50)	Nama
3.	URL	Varchar (100)	URL
4.	Message	Varchar (250)	Pesan
5.	Tanggal	Datetime	Tanggal
6.	Email	Varchar (30)	Email

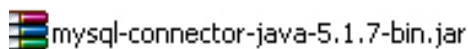
## 3.2 PERANCANGAN APLIKASI WEBSITE

### 3.2.1 Koneksi Database

Faktor penting dalam membangun sebuah aplikasi website dinamis adalah koneksi database. Dengan koneksi ini, kita bisa terhubung dengan database. Baik itu untuk mengambil *resource* dari database, maupun untuk mengolahnya.

Pada pembangunan website Wisata Kuliner ini, kami menggunakan bahasa pemrograman JSP (Java Server Pages) dengan kolaborasi MySQL sebagai databasenya. Untuk itu dibutuhkan suatu koneksi antara JSP dengan MySQL agar website bisa berjalan dengan baik.

Selain menggunakan scripting program, koneksi antara JSP dan MySQL juga membutuhkan suatu file library JSP yang biasa disebut sebagai "mysql connector java". Bentuk dari file library tersebut adalah:



**Gambar 3.7** file library JSP-MySQL connector

File ini diletakkan dalam folder /lib -nya Tomcat. File ini tidak otomatis ada saat penginstallan Tomcat, tetapi biasanya didownload dahulu dari internet.

Apabila file library "mysql connector java" sudah ada dalam folder /lib Tomcat, maka langkah selanjutnya adalah mendefinisikan koneksi JSP – MySQL menggunakan scripting program. Berikut ini contoh dari listing program koneksi JSP – MySQL: connect.jsp

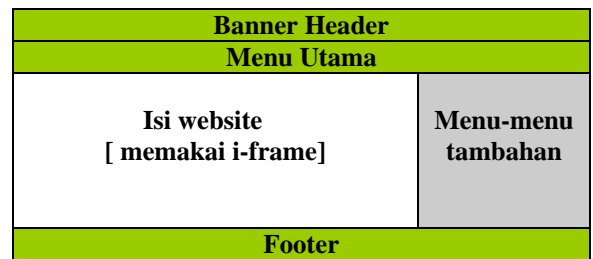
```
<%@ page import="java.sql.*" %>
<%
try{
    Class.forName("com.mysql.jdbc.Driver").newInstance();
}catch (Exception ex){
    out.println(ex.toString());
}
String Str =
"jdbc:mysql://127.0.0.1:3306/kuliner";
Connection conn = null;
String status="";
try{
    conn =
DriverManager.getConnection(Str,"root","");
    if (conn==null)
        status = "gagal";
    else
        status = "berhasil";
}catch (SQLException ex){
    out.println(ex.toString());
}
```

### 3.2.2 Desain Interface User

Dalam sistem ini akan dibuat aplikasi berbasis web. Untuk perancangan web digunakan program aplikasi Macromedia Dreamweaver MX dan bahasa pemrograman JSP (Java Server Page) dengan Database MySQL.

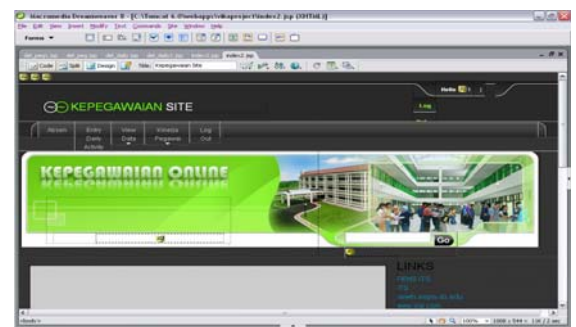
Halaman-halaman yang dibuat tidak semuanya di tampilkan pada browser. Sebagian halaman yang tidak ditampilkan digunakan untuk membantu jalannya proses sistem.

Berikut adalah gambar blok perencanaan halaman utama website kepegawaian :



**Gambar 3.12** Desain layout halaman utama website

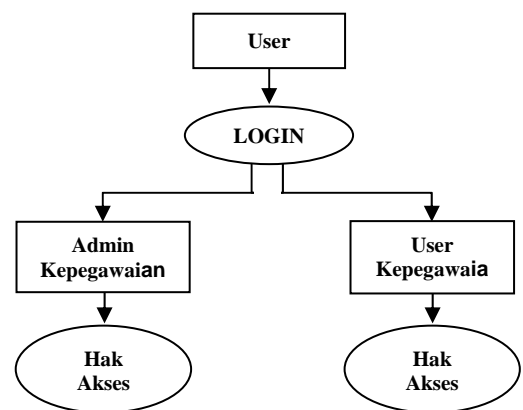
Untuk perancangan halaman website di Macromedia Dreamweaver seperti terlihat pada gambar di bawah ini :



**Gambar 3.13** Pembuatan halaman utama website pada Dreamweaver

### 3.2.3 Pendefinisian Hak Akses User

Pada sistem informasi ini terdapat 2 jenis login yang berbeda-beda, yang masing-masing memiliki hak akses yang berbeda-beda pula. Kedua jenis login beserta hak aksesnya bisa dilihat pada gambar di bawah ini:



**Gambar 3.14** Pendefinisian hak akses

User melakukan login pada website. Terdapat 2 macam mode login yaitu Admin website, admin kuliner dan member yang mempunyai level hak akses yang berbeda. Untuk hak akses level I mempunyai hak-hak antara lain :

- Melakukan input data pegawai, dp3
- Melakukan perubahan terhadap isi data pegawai serta penghapusan.



Untuk hak akses level II mempunyai kemampuan untuk :

- Melakukan absen serta input data daily activity
- Manage data sendiri

#### IV PENGUJIAN DAN ANALISA

##### 4.1 PENGUJIAN SISTEM

###### 4.1.1 Hasil Pengujian Sistem dan Analisa

Pengujian dilakukan untuk mengetahui apakah aplikasi yang telah dibangun telah berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan.

Dari hasil pengujian yang telah dilakukan dapat dilihat bahwa secara keseluruhan sistem informasi ini masih belum sempurna. Sistem informasi yang dibangun dengan menggunakan bahasa pemrograman JSP dan database MySQL ini dapat bekerja sesuai yang direncanakan untuk keperluan pemudahan dalam manage data kepegawaian.

Sistem ini terdiri dari USER dan ADMIN.

###### A. USER

User disini adalah pegawai. Masing-masing pegawai yang sudah terdaftar di PENS-ITS secara langsung akan mendapatkan hak user. Menu-menu pada user adalah sebagai berikut :

- A. Menu Absen  
Digunakan untuk absen masuk pegawai dan absen pulang.
- B. Menu Entry Daily Activity  
Digunakan untuk mengisikan daily activity-nya masing-masing pegawai.
- C. Menu Cek Daily Activity  
Digunakan pejabat atas untuk menyetujui daily activity pegawai
- D. Menu Profile  
Digunakan untuk melihat serta meng-edit profile sendiri
- E. Menu View Data
  - a. View Data Pegawai
  - b. View Absen
  - c. View Daily Activity
- F. Menu Kinerja Pegawai
  - a. Kinerja Absen
  - b. Kinerja Daily Activity
  - c. Kinerja DP3
- G. Menu Logout  
Digunakan untuk keluar dari sistem.

###### B. ADMIN

Untuk dapat meng-akses halaman admin maka harus login terlebih dahulu dan hanya admin saja yang dapat meng-akses halaman ini.

- A. Menu Entry Data
  - a. Entry Data Pegawai
  - b. Entry Data DP3
- B. Menu Edit dan Delete Data
  - a. Edit dan Delete Pegawai
  - b. Edit dan Delete Daily Activity
  - c. Edit dan Delete DP3
- C. Menu View
  - a. View Data Pegawai
  - b. View Absen

- c. View Daily Activity
- D. Menu Kinerja Pegawai
  - a. Kinerja Absen
  - b. Kinerja Daily Activity
  - c. Kinerja DP3
- E. Menu Logout  
Digunakan untuk keluar dari sistem.

Langkah-langkah pengujian pembuatan aplikasi analisa kinerja kepegawaian sebagai berikut :

1. Pastikan bahwa web server apache tomcat dan database mysql sudah berjalan dengan baik seperti yang dijelaskan pada bab 3.
2. Ketikkan <http://127.0.0.1:8080/vikaproject> atau <http://localhost/vikaproject> maka akan tampil halaman web pertama sebagai berikut :



Gambar 4.1 Preview Halaman Pertama

###### A. Menu Untuk USER

Untuk mengakses menu-menu didalamnya maka user harus login terlebih dahulu. Masing-masing pegawai yang telah terdaftar di PENS-ITS telah mendapatkan hak user dengan sendirinya. Isikan user dengan nama masing-masing dan passwordnya kita buat standar yaitu 123, maka user dapat mengakses halaman user.



Gambar 4.2 Halaman Login

Setelah login maka anda akan masuk pada halaman user seperti tampilan dibawah ini :



Gambar 4.3 Halaman User (Pegawai)



Pada halaman ini anda dapat melakukan absen, entry data daily activity dan melihat data-data tertentu yang ingin dikehendaki sesuai dengan hak akses.

Menu pertama yaitu menu absen. Diharapkan kepada pegawai untuk absen terlebih dahulu sebelum memulai aktifitasnya, dan absen pulang setelah waktunya pulang, dan terlihat seperti dibawah ini tampilannya :



**Gambar 4.4 Halaman Absen**

Apabila telah absen maka akan terdapat peringatan “Anda telah absen”, seperti tampilan dibawah ini :



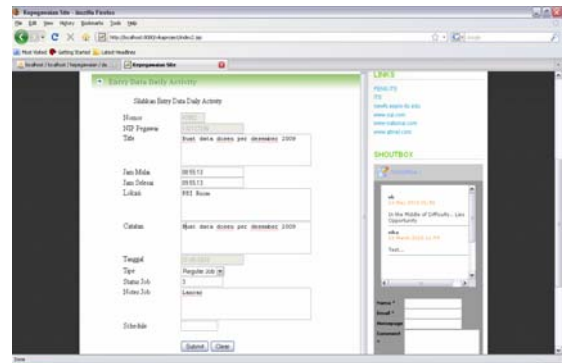
**Gambar 4.5 Halaman Absen Berhasil**

Setelah berhasil absen pada tanggal sekarang, maka absen masuk tombolnya tidak ada lagi aktif, dan terdapat tombol absen pulang pada menu absen pulang, seperti dibawah ini tampilannya :



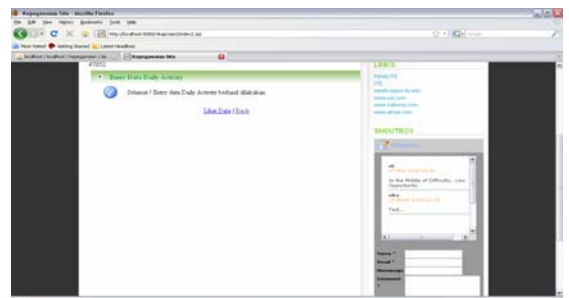
**Gambar 4.6 Halaman Absen Pulang**

Menu daily activity digunakan untuk mengisi aktifitas harian pegawai dengan cara pegawai meng-entrykan sendiri aktifitas masing-masing, dan berikut cara pengisiannya:



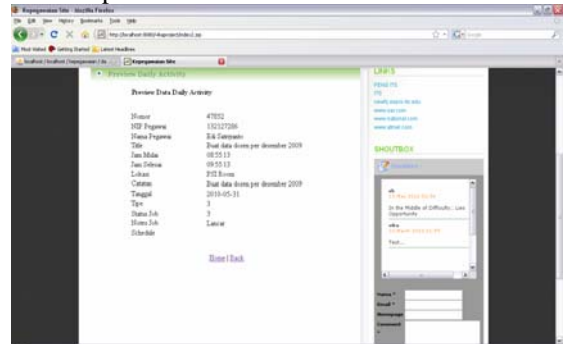
**Gambar 4.7 Halaman Entry Daily Activity**

Setelah Berhasil meng-entrykan data terdapat pemberitahuan bahwa entry data daily activity berhasil, berikut tampilannya :



**Gambar 4.8 Halaman Entry Daily Activity Berhasil**

Apabila ingin melihat data yang baru saja di inputkan dapat diklik pada “Lihat Data”, maka hasilnya akan terlihat seperti dibawah ini :



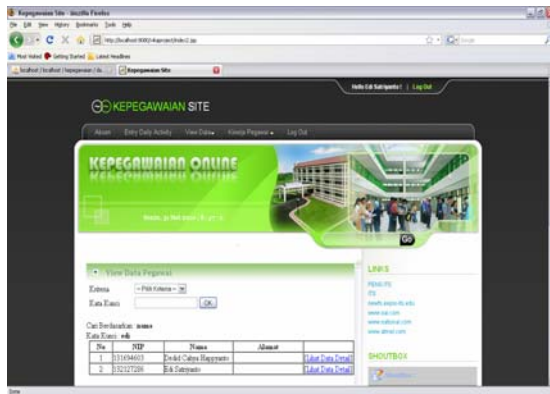
**Gambar 4.9 Halaman Preview Entry Daily Activity**

Menu selanjutnya view my profile yaitu bisa melihat profile pegawai yang login, serta bisa menng-edit datanya sendiri.



**Gambar 4.10 Halaman View Profile**

Menu view data digunakan untuk melihat data pegawai, absen, daily activity dan DP3. Disini menu view data pegawai adalah melihat data pegawai dengan kriteria yang di inginkan misal “nama”, dengan kata kunci yang di inginkan misal “edi”. Berikut tampilannya :



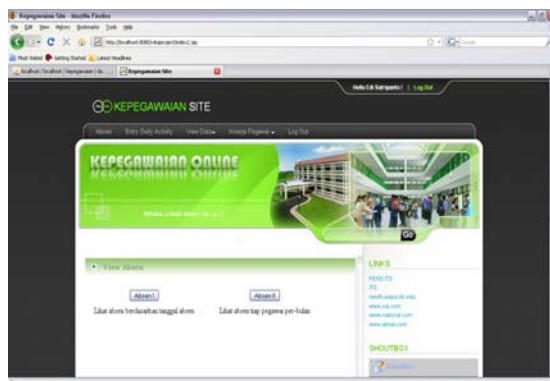
**Gambar 4.11 Halaman View Data Pegawai**

Untuk melihat detail datanya dapat di-klik pada “Lihat Data Detail”, maka akan terlihat seperti dibawah ini tampilannya :



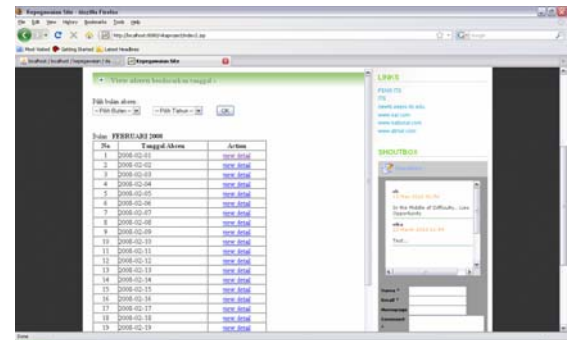
**Gambar 4.12 Halaman Detail Data Pegawai**

Untuk View yang selanjutnya adalah view absen yaitu melihat absen berdasarkan tanggal dan bulan, maka setelah dipilih bulan dan tahun yang dikehendaki akan tampil data absen sesuai dengan kriteria yang dipilih, seperti dibawah ini tampilannya :



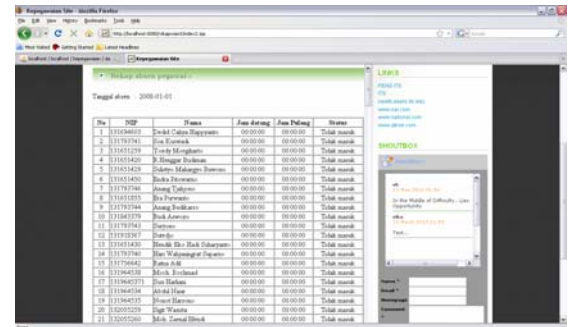
**4.13 Halaman Utama View Absen**

View absen pertanggal, jadi untuk melihatnya pilih bulan absen, maka akan muncul tanggal-tanggal absen seperti tampilan diibawah ini.



**Gambar 4.14 Halaman View Absen Per-tanggal**

Untuk melihat detail data absen per-tanggal dapat di-klik view detail, terlihat seperti dibawah ini tampilannya :



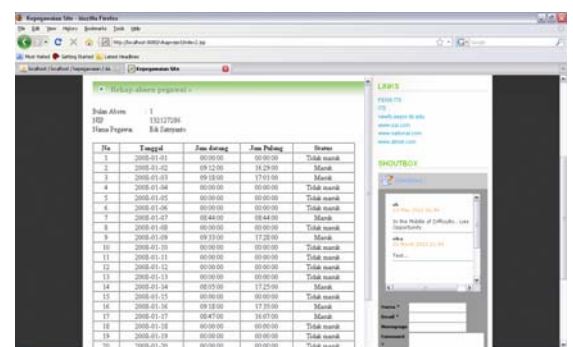
**Gambar 4.15 Halaman View Detail Absen Per-tanggal**

Untuk pilihan Absen II pada menu utama view absen akan tampil seperti dibawah ini. Setelah dipilih nama pegawai dan tahun absen, maka akan muncul data sesuai dengan kriteria yang dipilih.



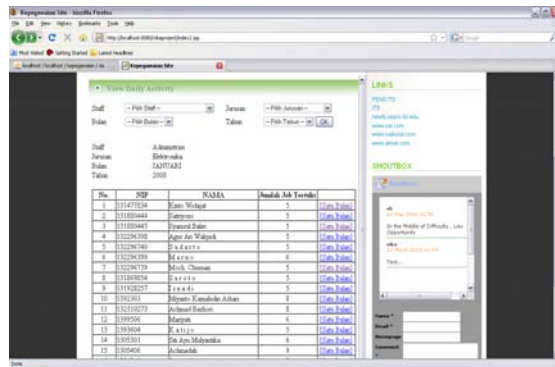
**4.16 Halaman View Absen Per-bulan**

Tampilan setelah di klik view detail pada form sebelumnya, akan menampilkan satu pegawai yang absen pada bulan yang dipilih sebelumnya.



**4.17 Halaman View Detail Absen Per-bulan**

View daily activity untuk melihat daily activity-nya pegawai sesuai dengan kriteria yang di pilih.



**4.18 Halaman View Daily Activity**

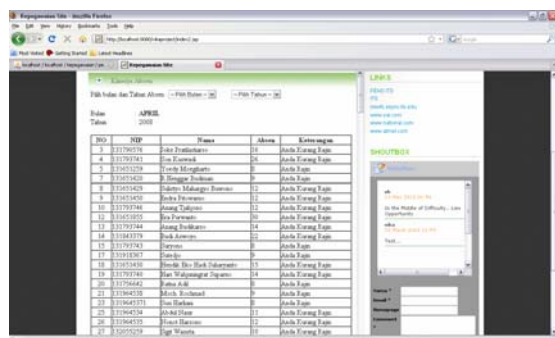
Tampilan setelah diklik “satu bulan” daily activity pegawai untuk melihat aktifitas masing-masing pegawai dalam satu bulan.



**4.19 Halaman View Detail Daily Activity**

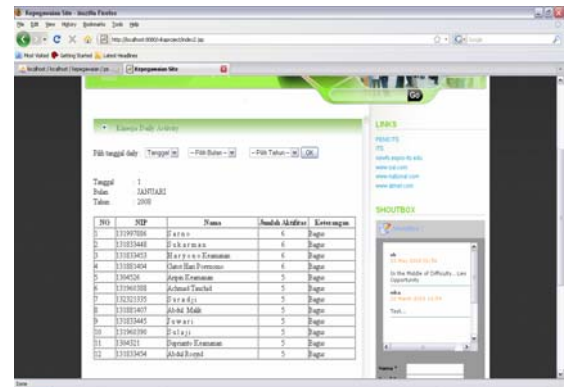
Menu kinerja pegawai digunakan untuk melihat hasil kinerja pegawai yaitu kinerja absen, daily activity dan DP3 yang telah dilakukan pegawai.

Kinerja absen yang dilakukan pegawai selama satu bulan dalam tahun tertentu. Apabila pegawai masuk kurang dari 10x maka dalam keterangan “Anda kurang Rajin”, dan pegawai yang masuknya lebih dari 10x keterangannya “Anda rajin, tingkatkan terus kedisiplinan anda”.



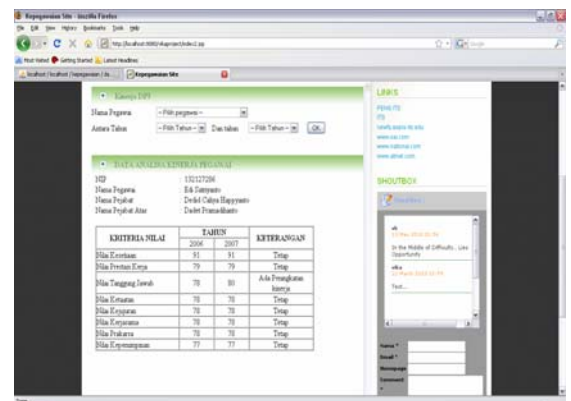
**Gambar 4.20 Halaman Kinerja Absen**

Kinerja daily activity dilakukan pegawai dalam 1 hari. Apabila pegawai melakukan aktifitas lebih dari 5x keterangannya “Mohon Ditingkatkan Lagi Aktivitasnya”, dan pegawai yang melakukan aktifitas kurang dari 5x dalam sehari keterangannya “Bagus, Tingkatkan Terus Daily Activity Anda”.



**Gambar 4.21 Halaman Kinerja Daily Activity**

Untuk kinerja DP3 yaitu membandingkan nilai antara tahun pertama dan kedua, apabila terdapat perbedaan nilai antara kedua tahun tersebut ada keterangan yang menunjukkan kepada nilai tersebut apakah naik, turun atau tetap, seperti terlihat dibawah ini tampilannya :



**Gambar 4.22 Halaman Kinerja DP3**

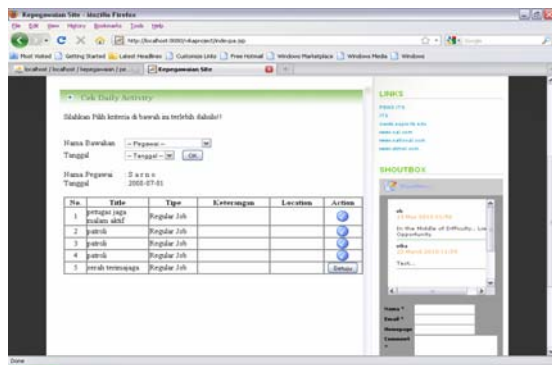
Menu berikutnya adalah menu logout, apabila user selesai mengoperasikan sistem, user bisa logout pada menu yang tersedia, berikut tampilannya :



**Gambar 4.23 Halaman Logout**

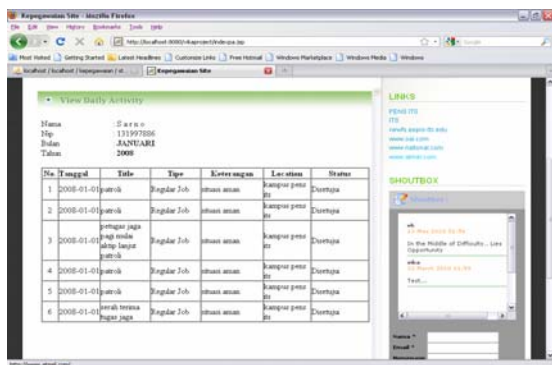
Pegawai ada yang menjadi pejabat atas yang membawahi beberapa bawahan. Pejabat tersebut mempunyai wewenang menyetujui daily activity pegawai. Maka setelah dia login mempunyai wewenang untuk menyetujui daily activity pegawai. Seperti ini tampilannya :





Gambar 4.24 Halaman Cek Daily Activity

Tampilan daily activity setelah disetujui dari atasan :



Gambar 4.25 Halaman Daily Activity Disetujui

## B. Menu Untuk ADMIN

Selanjutnya adalah menu Admin dimana halaman ini dikhususkan bagi admin yang bertugas untuk manage data pegawai, yaitu : entry data, meng-edit data serta menghapus data dan melihat hasil kinerja kepegawaian serta logout. Berikut akan ditampilkan saat user memilih menu ADMIN LOGIN, kemudian memasukkan username dan password, maka jika username dan password sesuai dengan data yang ada di database admin maka login akan berhasil. Jika tidak sesuai maka terdapat peringatan bahwa login error, untuk mengaksesnya anda dapat meng-klik menu pada pojok kanan atas yaitu Admin Login, maka akan masuk pada halaman seperti dibawah ini. Berikut tampilannya :



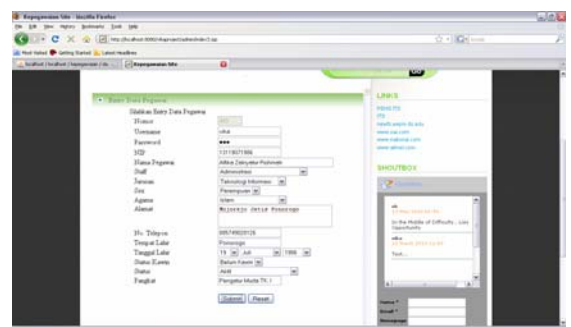
Gambar 4.26 Halaman Login Admin

Dari halaman login, jika proses login berhasil maka akan muncul halaman admin, dimana didalamnya terdapat menu entry data (terdapat entry pegawai dan entry dp3), edit dan delete data, view data pegawai, view kinerja pegawai dan logout.



Gambar 4.27 Halaman Admin

Untuk menu pertama dari halaman admin adalah menu entry data, yaitu untuk mengentry data Pegawai dan DP3. Berikut tampilan untuk entry data Pegawai :



Gambar 4.28 Halaman Entry Data Pegawai

Apabila entry data berhasil, akan terdapat pemberitahuan bahwa entry data berhasil, seperti dibawah ini tampilannya :



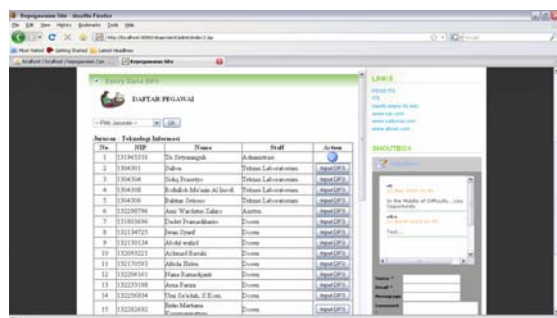
Gambar 4.29 Halaman Entry Data Berhasil

Apabila ingin meng-upload foto klik upload foto.



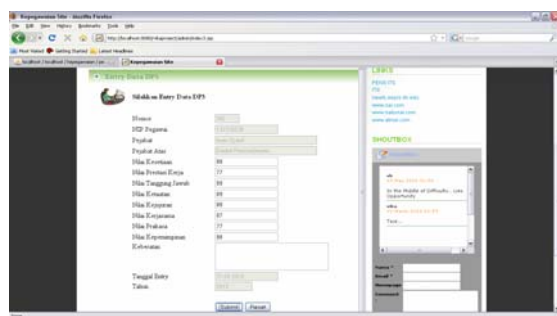
Gambar 4.30 Halaman Upload Foto

Entry DP3 dilakukan pertahun, apabila pada tahun ini belum di inputkan datanya, maka pada masing-masing pegawai terdapat tombol “Input DP3”, dan apabila data sudah di inputkan pada tahun ini akan terdapat tanda centang pada masing-masing pegawai. DP3 hanya untuk pegawai yang punya NIP.



**Gambar 4.31 Halaman Awal Input DP3**

Untuk meng-inputkan data klik tombol “Input DP3”, dan terlihat seperti dibawah ini input-an nya :



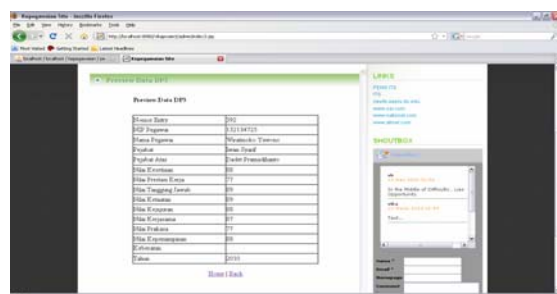
**Gambar 4.32 Halaman Input DP3**

Apabila berhasil meng-inputkannya akan ada pemberitahuan bahwa input data berhasil, seperti tampilan di bawah ini :



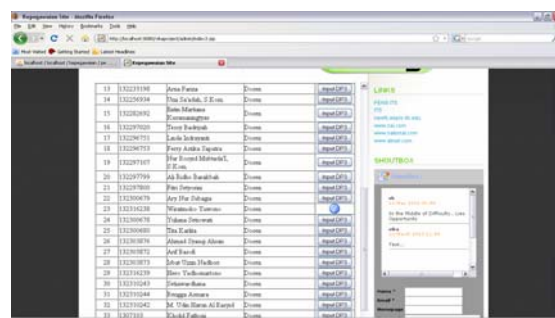
**Gambar 4.33 Halaman Input DP3 Berhasil**

Apabila ingin melihat data yang baru saja di inputkan klik lihat data.



**Gambar 4.34 Halaman Preview DP3**

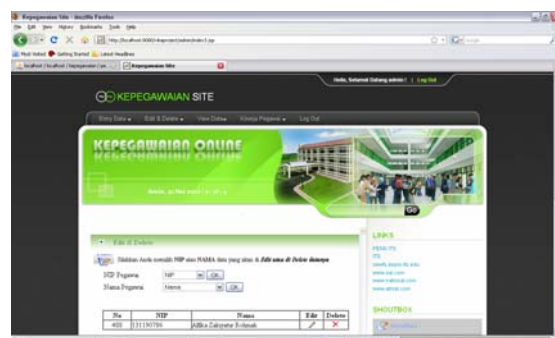
Setelah berhasil di inputkan, maka terdapat tanda centang pada pegawai tersebut.



**Gambar 4.35 Halaman Data DP3 telah di Inputkan**

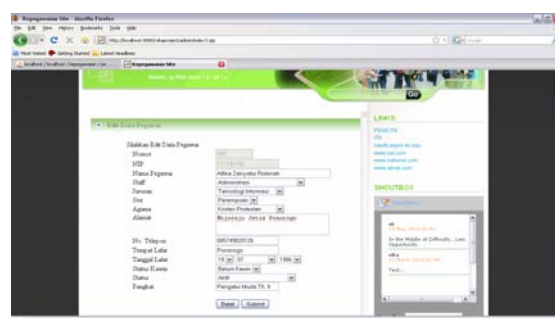
Menu edit dan delete digunakan oleh admin untuk meng-edit atau menghapus data mana saja yang ingin dikehendaki.

Untuk edit klik tanda “pensil”, maka akan masuk pada menu edit data :



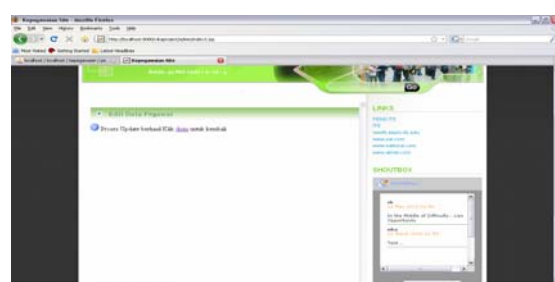
**Gambar 4.36 Halaman Edit dan Delete data Pegawai**

Pada form ini dapat di edit pada field apa saja yang di inginkan. Setelah selesai maka di submit dan akan ada pemberitahuan apakah data berhasil di edit atau tidak.



**Gambar 4.37 Halaman Edit Data Pegawai**

Pemberitahuan bahwa data berhasil di edit.



**Gambar 4.38 Halaman Edit Data Pegawai Berhasil**

Delete data sesuai dengan data mana yang akan di hapus dengan klik tanda silang seperti dibawah ini :



Gambar 4.39 Halaman Delete Data Pegawai

Kalau yakin ingin menghapus datanya klik tombol “Yakin”



Gambar 4.40 Halaman Konfirmasi Delete Data

Maka terdapat pemberitahuan data telah berhasil dihapus.



Gambar 4.41 Halaman Delete Berhasil

Menu view data digunakan untuk melihat data pegawai, absen, daily activity dan DP3. Disini menu view data pegawai adalah melihat data pegawai dengan kriteria yang di inginkan misal “nama”, dengan kata kunci yang di inginkan misal “edi”. Berikut tampilannya :



Gambar 4.42 Halaman View Data Pegawai

Untuk melihat detail datanya dapat di-klik pada “Lihat Data Detail”, maka akan terlihat seperti dibawah ini tampilanya :



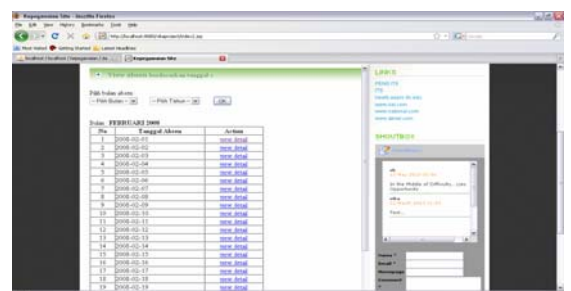
Gambar 4.43 Halaman Detail Data Pegawai

Untuk View yang selanjutnya adalah view absen yaitu melihat absen berdasarkan tanggal dan bulan, maka setelah dipilih bulan dan tahun yang dikehendaki akan tampil data absen sesuai dengan kriteria yang dipilih, seperti dibawah ini tampilannya :



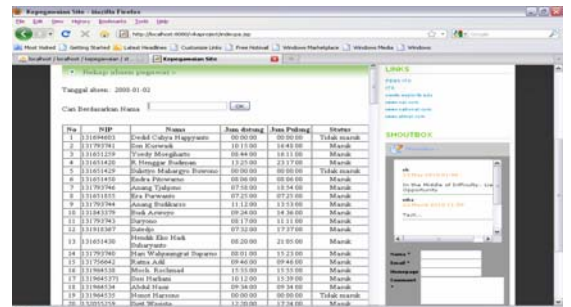
Gambar 4.44 Halaman Utama View Absen

View absen pertanggal, jadi untuk melihatnya pilih bulan absen, maka akan muncul tanggal-tanggal absen seperti tampilan dibawah ini.



Gambar 4.45 Halaman View Absen Per-tanggal

Untuk melihat detail data absen per-tanggal dapat di-klik view detail, terlihat seperti dibawah ini tampilannya :



Gambar 4.46 Halaman Detail Absen Per-tanggal

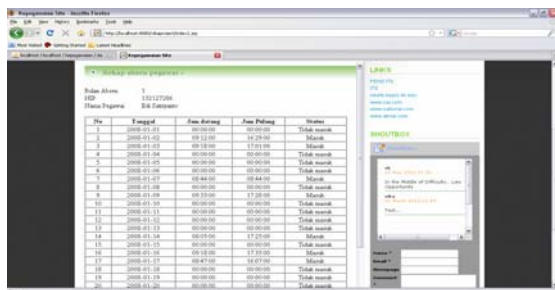


Untuk pilihan Absen II pada menu utama view absen akan tampil seperti dibawah ini. Setelah dipilih nama pegawai dan tahun absen, maka akan muncul data sesuai dengan kriteria yang dipilih.



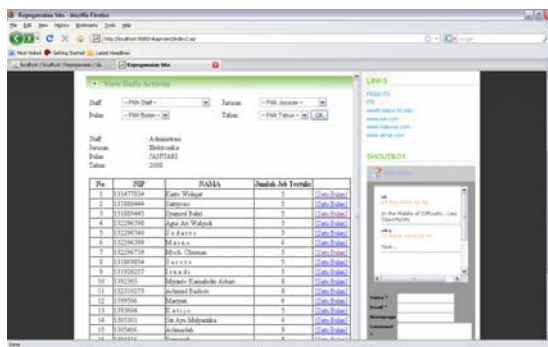
**Gambar 4.47 Halaman View Absen Per-bulan**

Tampilan setelah di klik view detail pada form sebelumnya, akan menampilkan satu pegawai yang absen pada bulan yang dipilih sebelumnya.



**Gambar 4.48 Halaman View Detail Absen Per-bulan**

View daily activity untuk melihat daily activity-nya pegawai sesuai dengan kriteria yang di pilih.



**Gambar 4.49 Halaman View Daily Activity**

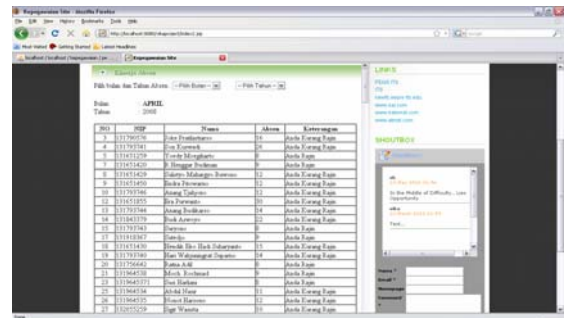
Tampilan setelah diklik “satu bulan” daily activity pegawai untuk melihat aktifitas masing-masing pegawai dalam satu bulan.



**Gambar 4.50 Halaman View Detail Daily Activity**

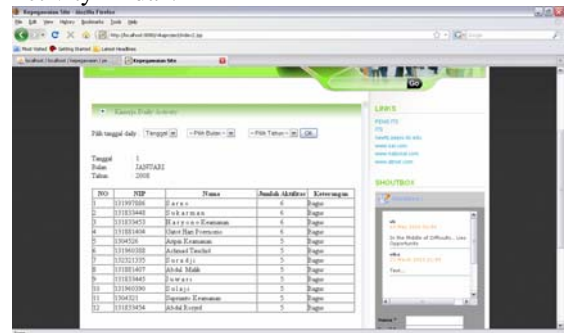
Menu kinerja pegawai digunakan untuk melihat hasil kinerja pegawai yaitu kinerja absen, daily activity dan DP3 yang telah dilakukan pegawai.

Kinerja absen yang dilakukan pegawai selama satu bulan dalam tahun tertentu. Apabila pegawai masuk kurang dari 10x maka dalam keterangan “Anda kurang Rajin”, dan pegawai yang masuknya lebih dari 10x keterangannya “ Anda rajin, tingkatkan terus kedisiplinan anda”.



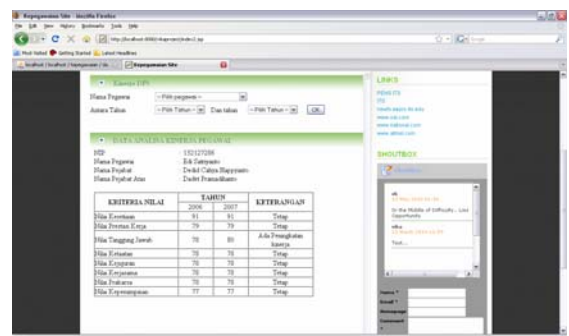
**Gambar 4.51 Halaman Kinerja Absen**

Kinerja daily activity dilakukan pegawai dalam 1 hari. Apabila pegawai melakukan aktifitas lebih dari 5x keterangannya “Mohon Ditingkatkan Lagi Aktivitasnya”, dan pegawai yang melakukan aktifitas kurang dari 5x dalam sehari keterangannya “Bagus, Tingkatkan Terus Daily Activity Anda”.



**Gambar 4.52 Halaman Kinerja Daily Activity**

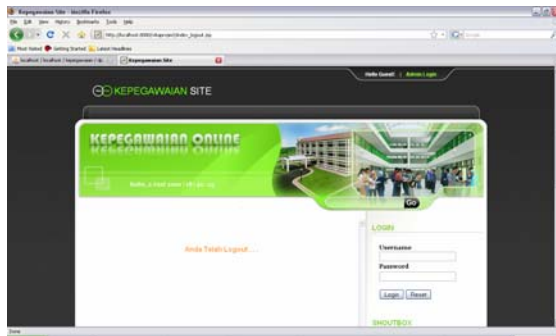
Untuk kinerja DP3 yaitu membandingkan nilai antara tahun pertama dan kedua, apabila terdapat perbedaan nilai antara kedua tahun tersebut ada keterangan yang menunjukkan kepada nilai tersebut apakah naik, turun atau tetap, seperti terlihat dibawah ini tampilannya :



**Gambar 4.53 Halaman Kinerja DP3**

Menu terakhir pada halaman admin adalah menu logout, dimana digunakan oleh admin untuk keluar dari

sistem, maka akan menuju halaman utama dari sistem ini. Berikut tampilannya :



Gambar 4.54 Halaman Logout

## V. PENUTUP

Pada bab-bab sebelumnya, yaitu dari bab I sampai dengan bab IV telah diuraikan beberapa hal yang berhubungan dengan pembuatan aplikasi ini, mulai dari latar belakang, teori-teori penunjang, perancangan dan pembuatan aplikasi, sampai dengan implementasinya yang disertai uji coba dan analisa. Pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari hasil-hasil pengujian aplikasi dan beberapa saran dengan harapan untuk lebih menyempurnakan perancangan yang telah dibuat.

### 5.1 Kesimpulan

Setelah dilakukan serangkaian pengujian terhadap sistem yang dibangun maka dapat diambil kesimpulan sebagai berikut :

- User harus login untuk masuk ke halaman selanjutnya
- Di dalam sistem ini juga disediakan halaman admin yang digunakan untuk manage data yang ada di database server, sehingga proses manage database lebih mudah dilakukan

### 5.2 Saran

Saran-saran yang dapat diambil dari kesimpulan diatas adalah sebagai berikut :

- Tampilan aplikasi ini dapat dikembangkan yang lebih baik lagi untuk memudahkan user dalam mengoperasikannya.
- Program ini dapat dikembangkan lebih lanjut dengan feature-feature yang lebih baik dan lebih praktis.
- Aplikasi yang dibuat pada proyek akhir ini masih perlu untuk disempurnakan lagi.

## DAFTAR PUSTAKA

- [1] Sano Alb. V. Dian, "24 Jam Menguasai HTML, JSP dan MySQL", CV. Andi Offset, Yogyakarta , 2005
- [2] Wasista, Sigit. 2002. *Pemrograman Web Diploma IV*. Surabaya
- [3] Wiyono, G. Sri Hartanti, Suharto, B.Herry, Wijono, M.Susilo, *Pemrograman Java Servlet dan JSP dengan NetBens*, CV. Andi Offset, Yogyakarta, 2006

- [4] Referensi bahasa pemrograman JSP : <http://www.jsp.net>
- [5] Referensi bahasa pemrograman JSP : <http://www.scripts.com>
- [6] Referensi bahasa pemrograman JSP : <http://www.ilmukomputer.com>
- [7] Referensi database MySQL : <http://www.mysql.com/>
- [8] Kadir, Abdul.2006.*Dasar Pemrograman Web Dinamis Dengan JSP*. Andi.